

# ΚΕΦΑΛΑΙΟ 5

## *Σχεδιασμός και υλοποίηση διαδικτυακών εφαρμογών*

### *Περιεχόμενα*

- 4.1 Εισαγωγή
- 4.2 Σχεδιασμός και Ενσωμάτωση Δικτύου
  - 4.2.1 Δίκτυα Υπολογιστών
  - 4.2.2 Τοπολογίες Εταιρικών Δικτύων (Internet, Intranet, Extranet)
  - 4.2.3 Ενσωμάτωση δικτύου
  - 4.2.4 Περιγραφή δικτύου
  - 4.2.5 Πρωτόκολλα επικοινωνίας
  - 4.2.6 Χωρητικότητα δικτύου
- 4.3 Το περιβάλλον ανάπτυξης και η αρχιτεκτονική της εφαρμογής
  - 4.3.1 Αρχιτεκτονική διαδικτυακού λογισμικού
  - 4.3.2 Τεχνολογίες διαδικτυακού λογισμικού
  - 4.3.3 Τεχνολογίες απεικόνισης
    - Εισαγωγή στην HTML και CSS
    - Εργαλεία HTML (notepad++)
    - Βασικές ετικέτες στην HTML
    - Λίστες και πίνακες στην HTML
    - Φόρμες στην HTML
  - 4.3.4 Τεχνολογίες προγραμματισμού πελάτη (client side)
    - Εισαγωγή στην Javascript
    - Μεταβλητές και πίνακες στην Javascript
    - Δομές επιλογής-επανάληψης στην Javascript
    - Αλληλεπίδραση Javascript με στοιχεία HTML (DOM)
    - Διαχείριση γεγονότων στην Javascript
    - Παράδειγμα ελέγχου ορθής συμπλήρωσης φόρμα με Javascript
- 4.4 Προγραμματισμός εξυπηρετητή σε PHP

- 4.4.1 Μεταβλητές
- 4.4.2 Δομή Επιλογής
- 4.4.3 Δομή Επανάληψης
- 4.4.4 Πίνακες
- 4.4.5 Συναρτήσεις
- 4.4.6 Πρόσβαση και διαχείριση Βάσεων Δεδομένων MySQL από το διαδίκτυο χρησιμοποιώντας PHP
- 4.4.7 Εισαγωγή Δεδομένων φόρμας σε Βάση Δεδομένων MySQL από το διαδίκτυο χρησιμοποιώντας PHP
- 4.4.8 MVC Πλαίσια για την δημιουργία διαδικτυακών PHP εφαρμογών (PHP Web Applications)

### ***Διδακτικοί στόχοι***

Στόχοι του κεφαλαίου αυτού είναι:

- Να εφαρμόζουν την διαδικασία σχεδιασμού εφαρμογών στο διαδίκτυο σύμφωνα με τις προτεινόμενες απαιτήσεις.
- Να περιγράφουν τα χαρακτηριστικά των περιβαλλόντων ανάπτυξης καθώς και τις αντίστοιχες αρχιτεκτονικές εφαρμογών.
- Να αναγνωρίζουν την διασύνδεση των τεχνολογιών για την ανάπτυξη διαδικτυακών εφαρμογών.
- Να εφαρμόζουν τις τεχνολογίες ανάπτυξης διαδικτυακών εφαρμογών για την υλοποίηση διαδικτυακών πληροφοριακών συστημάτων.
- Να αξιολογούν την δουλειά τους (προσωπική - ομαδική) με συγκεκριμένα κριτήρια αποδοτικότητας

## 5.1 Εισαγωγή

Όπως έχουμε δει σε προηγούμενες ενότητες, μπορούμε να φανταστούμε τον σχεδιασμό ενός συστήματος σαν ένα σύνολο από σχέδια που χρησιμοποιούνται για την κατασκευή ενός σπιτιού. Τα σχέδια οργανώνονται με βάση τα μέρη από τα οποία αποτελείται ένα σπίτι και περιγράφουν τα δωμάτια, τους τοίχους, τα παράθυρα, τις πόρτες, τις καλωδιώσεις, τις ηλεκτρολογικές εγκαταστάσεις καθώς και όλες τις υπόλοιπες λεπτομέρειες. Παρόμοια οργάνωση ακολουθούμε και στον σχεδιασμό ενός συστήματος παρόλο που τα σημεία που περιγράφουμε διαφέρουν από αυτά ενός σπιτιού. Σχεδιάζουμε δηλαδή και καθορίζουμε τα διάφορα συστατικά του συστήματος. Το αποτέλεσμα της σχεδίασης αποτελεί ένα προσχέδιο για την μετέπειτα υλοποίηση του.

Η διαδικασία του σχεδιασμού αποτελείται από δύο επίπεδα. Ξεκινάει από ένα γενικό σχεδιασμό, τον λεγόμενο αρχιτεκτονικό σχεδιασμό (architectural design) και καταλήγει σε ένα πιο ειδικό σχεδιασμό στον λεπτομερή σχεδιασμό (detail design). Κατά την διάρκεια του αρχιτεκτονικού σχεδιασμού περιγράφουμε τα σημεία από τα οποία αποτελείται το σύστημα. Προσδιορίζουμε τη συνολική δομή και μορφή της λύσης σε ένα πιο αφηρημένο επίπεδο. Κατά τον λεπτομερή σχεδιασμό εστιάζουμε στις λεπτομέρειες, δηλαδή στο περιβάλλον ανάπτυξης και στις διάφορες τεχνολογίες υλικού και λογισμικού που θα χρησιμοποιηθούν.

Ο σχεδιασμός ενός πληροφοριακού συστήματος περιλαμβάνει μεταξύ άλλων τον σχεδιασμό και την ενσωμάτωση του δικτύου καθώς και τον σχεδιασμό της αρχιτεκτονικής της εφαρμογής και του λογισμικού. Δύο είναι τα βασικά ερωτήματα που πρέπει να απαντηθούν με την ολοκλήρωση των δύο παραπάνω σχεδιαστικών δραστηριοτήτων.

Το πρώτο ερώτημα αφορά την σχεδίαση και την ενσωμάτωση της δικτυακής δραστηριότητας: Έχουν καθοριστεί λεπτομερώς πώς τα διάφορα μέρη του συστήματος θα επικοινωνούν μεταξύ τους; Οι χρήστες ενός πληροφοριακού συστήματος εκτελούν μία σειρά από ενέργειες από διαφορετικές τοποθεσίες. Οι τοποθεσίες αυτές πρέπει να είναι συνδεδεμένες μεταξύ τους με τέτοιο τρόπο ώστε να ικανοποιούν με τον καλύτερο δυνατό τρόπο της απαιτήσεις της επιχείρησης ή του οργανισμού. Ο τομέας της επικοινωνίας της επιχείρησης ή του οργανισμού μέσω ενός δικτύου υπολογιστών αποτελεί αντικείμενο μελέτης της σχεδίασης του δικτύου. Κατά την σχεδίαση του δικτύου καθορίζονται με λεπτομέρεια η μορφή του δικτύου, οι δυνατότητες του, το υλικό που θα χρησιμοποιηθεί καθώς και σημαντικά τεχνικά ζητήματα όπως η αξιοπιστία, η ασφάλεια, η απόδοση και ο συγχρονισμός.

Το δεύτερο βασικό ερώτημα που πρέπει να απαντηθεί με την ολοκλήρωση της σχεδίασης της αρχιτεκτονικής της εφαρμογής και του λογισμικού είναι το εξής: Έχουν καθοριστεί οι χρησιμοποιούμενες τεχνολογίες λογισμικού που θα υποστηρίξουν τις δραστηριότητες του συστήματος που διενεργούνται από τους χρήστες και τους υπολογιστές; Για παράδειγμα, κατά την διαδικασία της ανάλυσης

αποφασίζεται αν οι χρήστες θα είναι σε θέση να έχουν πρόσβαση στο νέο σύστημα μόνο στους χώρους εργασίας τους ή θα έχουν επίσης την δυνατότητα να εργάζονται από το σπίτι τους μέσω μιας σύνδεσης στο Internet. Αν είναι επίσης απαραίτητο να επιτραπεί σε απομακρυσμένες ασύρματες συσκευές να συνδεθούν με το σύστημα. Ποια είναι τα διάφορα είδη δραστηριοτήτων καθώς και ο όγκος των δραστηριοτήτων αυτών που το νέο σύστημα θα πρέπει να είναι σε θέση να χειριστεί. Η υποστήριξη των παραπάνω εργασιών των χρηστών από τεχνολογικής πλευράς αποτελεί αντικείμενο μελέτης της σχεδίασης της αρχιτεκτονικής της εφαρμογής και του λογισμικού. Για παράδειγμα, κατά την σχεδίαση της αρχιτεκτονικής της εφαρμογής θα πρέπει να αποφασιστεί αν θα χρησιμοποιηθούν υπηρεσίες διαδικτύου και στην περίπτωση που η απάντηση είναι θετική ποια θα είναι η χρησιμοποιούμενη υποδομή καθώς και τα αντίστοιχα εργαλεία ανάπτυξης.

Στη συνέχεια θα αναφερθούμε διεξοδικά στα διάφορα μέρη που αποτελούν τον σχεδιασμό του δικτύου καθώς και της αρχιτεκτονικής της εφαρμογής. Η παρουσίαση των θεματικών ενοτήτων του κεφαλαίου συμπληρώνεται στο τέλος με εκτενή αναφορά στα εργαλεία ανάπτυξης διαδικτυακών εφαρμογών.

## 5.2 Σχεδιασμός και Ενσωμάτωση Δικτύου

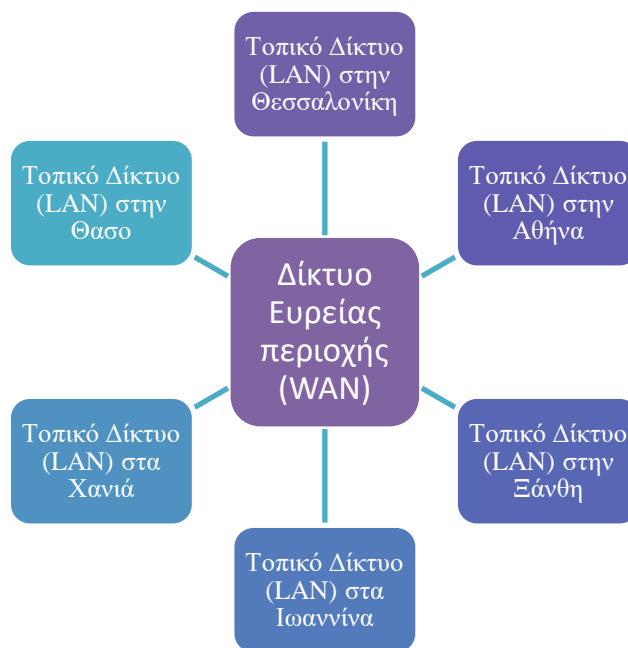
Τα **δίκτυα υπολογιστών** χρησιμοποιούνται για τις διάφορες δραστηριότητες των επιχειρήσεων και των οργανισμών. Υποστηρίζουν δυνατότητα επικοινωνίας μεταξύ των υπολογιστικών συστημάτων και των χρηστών. Η δυνατότητα αυτή μπορεί να αφορά παροχή διαφόρων υπηρεσιών (όπως για παράδειγμα τηλεφωνική υπηρεσία, video conferencing, e-mail) καθώς και κατανομή ψηφιακών πόρων (όπως για παράδειγμα πρόσβαση σε ηλεκτρονικά έγγραφα, σε εφαρμογές και βάσεις δεδομένων). Για τον λόγο αυτόν, τα σύγχρονα έργα ανάπτυξης πληροφοριακών συστημάτων συμπεριλαμβάνουν στον σχεδιασμό τους και αυτόν του δικτύου.

Ο σχεδιασμός του δικτύου είναι ένα σημαντικό ζήτημα που πρέπει να αντιμετωπιστεί νωρίς κατά τον σχεδιασμό ενός οποιουδήποτε πολυεπίπεδου (multitiered) συστήματος. Τα βασικά θέματα του σχεδιασμού του δικτύου ενός πληροφοριακού συστήματος αφορούν τα παρακάτω:

- Ενσωμάτωση των αναγκών του δικτύου του νέου συστήματος στην υπάρχουσα διαδικτυακή υποδομή
- Περιγραφή των δραστηριοτήτων επεξεργασίας και σύνδεσης στο δίκτυο από κάθε θέση του συστήματος
- Περιγραφή των πρωτόκολλων επικοινωνίας και του ενδιάμεσου λογισμικού (middleware) που συνδέει τα διάφορα στρώματα
- Διασφάλιση της επαρκούς χωρητικότητας του δικτύου

### 5.2.1 Δίκτυα Υπολογιστών

Στο παρακάτω σχήμα 5.1 παρουσιάζεται ένα πιθανό δίκτυο υπολογιστών μιας επιχείρησης. Κάθε τοπικό δίκτυο LAN εξυπηρετεί μία γεωγραφική τοποθεσία και όλα τα τοπικά δίκτυα συνδέονται μέσω ενός δικτύου ευρείας περιοχής WAN. Οι χρήστες και οι υπολογιστές μιας συγκεκριμένης τοποθεσίας επικοινωνούν μεταξύ τους μέσω του τοπικού τους δικτύου LAN. Η επικοινωνία μεταξύ χρηστών που βρίσκονται σε διαφορετικές γεωγραφικές τοποθεσίες διεξάγεται μέσω των τοπικών τους δικτύων αντίστοιχα καθώς και του δικτύου ευρείας περιοχής που ενώνει τα τοπικά δίκτυα της εταιρίας ή του οργανισμού μεταξύ τους. Ο δρομολογητής συνδέει κάθε τοπικό δίκτυο με το δίκτυο ευρείας περιοχής. Πακέτα πληροφοριών ενός χρήστη ή ενός υπολογιστή ενός τοπικού δικτύου Α που προορίζονται για χρήστες ή υπολογιστές ενός άλλου τοπικού δικτύου Β μεταφέρονται μέσω των δρομολογητών στο δίκτυο ευρείας περιοχής και από εκεί στον αντίστοιχο χρήστη ή υπολογιστή του τοπικού δικτύου Β. Διάφορες τεχνολογίες χρησιμοποιούνται για την υλοποίηση των τοπικών και ευρείας περιοχής δικτύων, όπως τεχνολογίες Ethernet και μισθωμένες γραμμές ψηφιακής τεχνολογίας αντίστοιχα. Αυτές ποικίλουν σύμφωνα με το κόστος, την χωρητικότητα και την αξιοπιστία.



**Σχήμα 5.1:** Δίκτυο υπολογιστών μιας επιχείρησης ή ενός οργανισμού

Ένα δίκτυο υπολογιστών με τον κατάλληλο εξοπλισμό και επαρκή δυνατότητα μεταφοράς δεδομένων μπορεί να υποστηρίξει ταυτόχρονα πολλές υπηρεσίες. Υπάρχουν πολλοί τρόποι διανομής πόρων ενός πληροφοριακού συστήματος σε ένα δίκτυο υπολογιστών. Χρήστες, εφαρμογές και βάσεις δεδομένων μπορούν να είναι αξιοποιούν τις δυνατότητες ενός υπολογιστικού συστήματος ενός τοπικού δικτύου, διαφορετικών υπολογιστικών συστημάτων του ίδιου τοπικού δικτύου ή διαφορετικών υπολογιστικών συστημάτων διαφόρων τοπικών δικτύων. Τμήματα εφαρμογών και βάσεων δεδομένων κατανέμονται σε διάφορα υπολογιστικά συστήματα σε όλο το εύρος του δικτύου υπολογιστών μιας επιχείρησης ή ενός οργανισμού.

### 5.2.2 Τοπολογίες Εταιρικών Δικτύων (Internet, Intranet, Extranet)

Το **Διαδίκτυο (Internet)** είναι ένα παγκόσμιο σύστημα δικτύων υπολογιστών που συνδέονται μεταξύ τους χρησιμοποιώντας ένα κοινό χαμηλού επίπεδου πρότυπο δικτύωσης που ονομάζεται TCP/IP. Ο Παγκόσμιος Πληροφοριακός Ιστός (World Wide Web ή Web) είναι ένα σύνολο από πόρους (προγράμματα, αρχεία και υπηρεσίες) που μπορούν να προσπελαστούν μέσω του Διαδικτύου μέσω συγκεκριμένων πρωτοκόλλων. Το Διαδίκτυο είναι η υποδομή πάνω στην οποία βασίζεται ο Παγκόσμιος Ιστός. Με άλλα λόγια, οι πόροι του Παγκόσμιου Ιστού παρέχονται στους χρήστες μέσω του Διαδικτύου.

Ένα **εσωτερικό εταιρικό δίκτυο (intranet)** είναι ένα ιδιωτικό δίκτυο υπολογιστών ή ένα κλειστό εταιρικό δίκτυο που είναι προσπελάσιμο μόνο από ένα περιορισμένο σύνολο χρηστών. Οι χρήστες αυτοί είναι συνήθως εξουσιοδοτημένα μέλη της ίδιας εταιρείας ή της ίδιας ομάδας εργασίας και έχουν δικαιώματα πρόσβασης σε συγκεκριμένους μόνο πόρους. Τα μη δημοσιευμένα ονόματα πόρων, τα firewalls και οι κωδικοί πρόσβασης ατόμων ή ομάδων προστατεύουν τα intranet από μη-εξουσιοδοτημένους εξωτερικούς εισβολείς.

Το **διευρυμένο εταιρικό δίκτυο (extranet)** είναι ένα διευρυμένο εσωτερικό εταιρικό δίκτυο (intranet) το οποίο έχει επεκταθεί ώστε να περιλαμβάνει τους εξουσιοδοτημένους εξωτερικούς συνεργάτες μιας επιχείρησης ή ενός οργανισμού όπως για παράδειγμα είναι οι προμηθευτές, οι μεγάλοι πελάτες και οι στρατηγικοί εταίροι. Το διευρυμένο εταιρικό δίκτυο (extranet) επιτρέπει σε εξουσιοδοτημένες ομάδες συνεργαζόμενων εταιρειών να ανταλλάσσουν πληροφορίες και να συντονίζουν τις δραστηριότητές τους σχηματίζοντας κατά αυτόν τον τρόπο ένα εικονικό οργανισμό (**virtual organization**). Μια ευρέως χρησιμοποιούμενη μέθοδος εφαρμογής ενός διευρυμένου εταιρικού δικτύου (extranet) είναι αυτή του εικονικού ιδιωτικού δικτύου (**virtual private network ή VPN**), το οποίο είναι ένα ιδιωτικό δίκτυο που είναι ασφαλές και προσβάσιμο μόνο στα μέλη του οργανισμού (ή του εικονικού οργανισμού).

### 5.2.3 Ενσωμάτωση δικτύου

Οι σύγχρονες επιχειρήσεις και οργανισμοί βασίζονται σε δίκτυα υπολογιστών για να υποστηρίξουν τις πολλές και διαφορετικές εφαρμογές τους. Στις περιπτώσεις που το υφιστάμενο δίκτυο μπορεί να φιλοξενήσει το νέο σύστημα, η ενσωμάτωση του νέου πληροφοριακού συστήματος πραγματοποιείται με τέτοιο τρόπο ώστε να μην επηρεάζονται οι υπάρχουσες εφαρμογές. Συνήθως απαιτούνται ελάχιστες αλλαγές όπως η προσθήκη συνδέσεων για νέους εξυπηρετητές, η τροποποίηση της δρομολόγησης και η διαμόρφωση του τοίχου προστασίας (firewall) για να μπορέσει να καταστεί δυνατή η επικοινωνία μεταξύ των νέων επιπέδων εφαρμογών. Ο προγραμματισμός επιπρόσθετων σημαντικών αλλαγών, όπως αναβάθμιση της χωρητικότητας, εισαγωγή νέων πρωτοκόλλων επικοινωνίας ή τροποποίηση πρωτοκόλλων ασφάλειας είναι πολύ πιο περίπλοκος και αποτελεί ευθύνη του

διαχειριστή του δικτύου, διότι είναι ο μόνος που γνωρίζει το υφιστάμενο δίκτυο και τον τρόπο που οι διάφορες διαδικτυακές εφαρμογές εκτελούνται και συνεργάζονται μεταξύ τους. Ο ρόλος του αναλυτή είναι να παρέχει στον διαχειριστή του δικτύου αρκετές πληροφορίες και χρόνο έτσι ώστε να καταστεί δυνατή η ανάπτυξη, η δοκιμή και η ολοκλήρωση του νέου συστήματος.

#### **5.2.4 Περιγραφή δικτύου**

Υπάρχουν αρκετοί τρόποι περιγραφής της διαδικτυακής υποδομής μιας εφαρμογής. Συνήθως ένα διάγραμμα δικτύου περιγράφει το πώς τα διάφορα επίπεδα εφαρμογών κατανέμονται στις διάφορες τοποθεσίες και υπολογιστικά συστήματα. Πιο συγκεκριμένα, αποτυπώνονται στο διάγραμμα, που «τρέχουν» τα διάφορα προγράμματα εφαρμογών, που βρίσκονται οι εξυπηρετητές και οι σταθμοί εργασίας και πώς είναι οργανωμένοι οι διαδικτυακοί πόροι. Οι απαιτήσεις της εφαρμογής και η πολιτική του οργανισμού ή της εταιρείας είναι αυτές που καθορίζουν την θέση των εξυπηρετητών, τις διόδους επικοινωνίας και τα ζητήματα ασφάλειας.

#### **5.2.5 Πρωτόκολλα επικοινωνίας**

Με βάση το διάγραμμα του δικτύου προσδιορίζονται οι απαιτήσεις στα πρωτόκολλα επικοινωνίας, δηλαδή η μορφή και η σειρά των μηνυμάτων που ανταλλάσσονται μεταξύ δύο ή περισσότερων κόμβων που επικοινωνούν μεταξύ τους καθώς και οι διάφορες ενέργειες που διενεργούνται κατά τη μετάδοση ή τη λήψη ενός μηνύματος.

Για παράδειγμα, στην περίπτωση που έχουμε ένα ιδιωτικό δίκτυο δικτύου ευρείας περιοχής WAN ποια πρωτόκολλα επικοινωνίας πρέπει να υποστηρίζονται για να μπορούν οι διάφοροι εξουσιοδοτημένοι χρήστες να συνδεθούν και να κάνουν ερωτήματα; Και στην περίπτωση κατάρρευσης του ιδιωτικού αυτού WAN θα μπορούν τα παραπάνω πρωτόκολλα επικοινωνίας να υποστηρίξουν εναλλακτικές λύσεις, όπως αυτής της δρομολόγησης των μηνυμάτων μέσω κρυπτογραφημένων συνδέσεων; Όλοι οι πελάτες θα πρέπει να είναι σε θέση να στείλουν αιτήματα HTTP και να λαμβάνουν ενεργό περιεχόμενο ως HTML φόρμες και ενσωματωμένα σενάρια. Οι διακομιστές εφαρμογών θα πρέπει να είναι σε θέση να επικοινωνούν με υπηρεσίες πιστωτικού έλεγχου μέσω του Διαδικτύου. Οι τοίχοι προστασίας (Firewalls) και οι δρομολογητές πρέπει να ρυθμιστούν για να υποστηρίξουν όλες τις αλληλεπιδράσεις μεταξύ των σταθμών εργασίας, των υπολογιστών των πελατών, των εξυπηρετητών Web/εφαρμογών καθώς και των άλλων υπηρεσιών. Ένα τοπικό δίκτυο κέντρο δεδομένων πρέπει να υποστηρίζει τουλάχιστον ένα πρωτόκολλο για τη μετάδοση ερωτημάτων βάσεων δεδομένων και απαντήσεων μεταξύ των εξυπηρετητών κεντρικών υπολογιστών (mainframe) και Web/εφαρμογών.

#### **5.2.6 Χωρητικότητα δικτύου**

Κατά την διάρκεια της ανάλυσης καταγράφονται πληροφορίες σχετικά με τις δραστηριότητες και το είδος των δραστηριοτήτων ανά τοποθεσία. Δηλαδή, ποιες

οντότητες δεδομένων προσπελούνται (για παράδειγμα πελάτης, παραγγελία), στο πλαίσιο ποιών δραστηριοτήτων (για παράδειγμα δημιουργία νέας παραγγελίας), ποιες ενέργειες γίνονται πάνω σε αυτά (για παράδειγμα δημιουργία, ενημέρωση, διάβασμα, διαγραφή) και που συμβαίνουν όλα τα παραπάνω (δηλαδή σε ποιόν εξυπηρετητή συγκεκριμένα). Οι πληροφορίες αυτές μπορούν να χρησιμεύσουν για μία πρώτη εκτίμηση των απαιτήσεων για ικανότητα επικοινωνίας. Για παράδειγμα, για την δημιουργία μίας νέας παραγγελίας, ο όγκος των δεδομένων που θα μεταδοθούν είναι της τάξης των 500 bytes και ο μέσος χρόνος προσπέλασης των δεδομένων θα είναι της τάξης των 2 προσπελάσεων ανά λεπτό και σε ώρες αιχμής θα είναι 10 προσπελάσεις ανά λεπτό.

Με τον σχεδιασμό και την ολοκλήρωση των περισσότερων τμημάτων του πληροφοριακού συστήματος, όπως αυτής των βάσεων δεδομένων είναι δυνατό να έχουμε μία καλύτερη προσέγγιση των απαιτήσεων σε ικανότητα επικοινωνίας ή να μπορούμε να μετρήσουμε με περισσότερη ακρίβεια τον πραγματικό χρόνο μετάδοσης δεδομένων.

Το επόμενο βήμα στον σχεδιασμό του πληροφοριακού συστήματος αφορά την αρχιτεκτονική της εφαρμογής καθώς και το αντίστοιχο λογισμικό εφαρμογών. Οι αποφάσεις που λαμβάνονται κατά τον σχεδιασμό της αρχιτεκτονικής των εφαρμογών επηρεάζουν αλλά και επηρεάζονται με την σειρά τους από την υφιστάμενη διαδικτυακή εφαρμογή. Στην παρακάτω ενότητα θα παρουσιαστούν οι διάφορες επιλογές που είναι δυνατό να συναντήσουμε στην αρχιτεκτονική μιας εφαρμογής καθώς και στο αντίστοιχο λογισμικό.

### 5.3 Το περιβάλλον ανάπτυξης και η αρχιτεκτονική της εφαρμογής

Ο **παγκόσμιος ιστός (World Wide Web)** εξαπλώνεται διαρκώς σε ολόένα και περισσότερους τομείς της ανθρώπινης δραστηριότητας. Αποτελεί τομέα των Τεχνολογιών Πληροφορικής και Επικοινωνιών που εξελίσσεται ραγδαία. Η εξέλιξη του παγκόσμιου ιστού οφείλεται στην βελτίωση των υποδομών δικτύου, στην μείωση του κόστους πρόσβασης στο διαδίκτυο για τον μέσο χρήστη καθώς στην ευκολία πρόσβασης στις υπηρεσίες που προσφέρει από οποιοδήποτε μέρος του κόσμου. Αν και μάθαμε σε προηγούμενες τάξεις ποιες είναι αυτές οι υπηρεσίες- δυνατότητες που προσφέρει ο παγκόσμιος ιστός, δεν μπορούμε να αποκλείσουμε νέες καινοτόμες υπηρεσίες που δημιουργούνται σε καθημερινή βάση. Γι αυτό η μορφή του παγκόσμιου ιστού έχει αλλάξει ριζικά από την εποχή που δημιουργήθηκε από τον **Tim Berners-Lee το 1989**. Η μελλοντική του μορφή δεν είναι καθόλου εύκολα προβλέψιμη.

Η ταχύτατη ανάπτυξη του παγκόσμιου ιστού οφείλεται και στο γεγονός ότι πολλοί φορείς (κράτη, εταιρείες, κοινότητες προγραμματιστών, ιδιώτες) επενδύουν χρόνο και χρήμα σε αυτόν. Το γεγονός αυτό όμως είχε ως αποτέλεσμα την δημιουργία πολλών εργαλείων, προτύπων και γενικότερα τεχνολογιών για την ανάπτυξη διαδικτυακών εφαρμογών. Τάση η οποία θα συνεχιστεί και στο μέλλον. Γι αυτό το λόγο ο



μελλοντικός προγραμματιστής διαδικτυακών εφαρμογών θα πρέπει να έχει την ικανότητα γρήγορης παρακολούθησης και μάθησης νέων γνώσεων-τεχνολογιών.

Σκοπός αυτής της ενότητας είναι να κατανοήσουμε και να μάθουμε τις βασικές τεχνολογίες ανάπτυξης –υλοποίησης διαδικτυακών εφαρμογών με τέτοιο τρόπο ώστε να μπορούμε στο μέλλον να προσαρμοζόμαστε γρήγορα και εύκολα μόνοι μας στις τεχνολογικές εξελίξεις του χώρου και γιατί όχι να διαμορφώσουμε εμείς την μελλοντική εξέλιξη του παγκόσμιου ιστού.

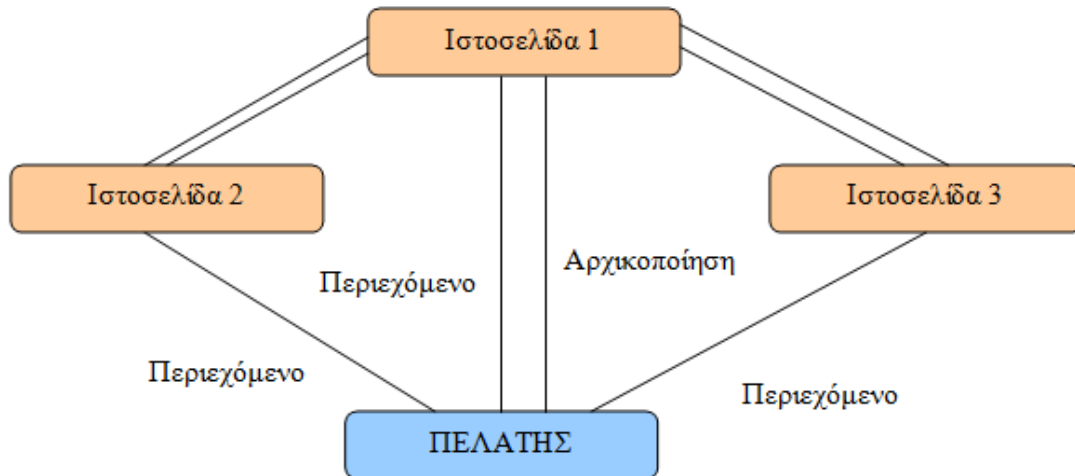
### 5.3.1 Αρχιτεκτονική διαδικτυακού λογισμικού

Όπως όλα τα ανθρώπινα πολύπλοκα επινοήματα (κτίρια, ηλεκτρονικές συσκευές, αυτοκίνητα κ.α.) ή καλύτερα εφευρέσεις που αποτελούνται από πολλά διαφορετικά μέρη που συνεργάζονται μεταξύ τους έτσι και το διαδικτυακό λογισμικό χαρακτηρίζεται από την αρχιτεκτονική του. Όπως ήδη έχουμε μάθει από προηγούμενα κεφάλαια με τον όρο αρχιτεκτονική αναφερόμαστε στα συστατικά που αποτελούν το διαδικτυακό λογισμικό καθώς και στην δομή με την οποία αλληλεπιδρούν. Πριν δοθεί η γενική αρχιτεκτονική διαδικτυακού λογισμικού κρίνεται σκόπιμο να καταλάβουμε τι είναι διαδικτυακό λογισμικό – εφαρμογή και πως αυτό διαφέρει από μια απλή ιστοσελίδα.

**Ορισμός:** Διαδικτυακή εφαρμογή-λογισμικό είναι μια εφαρμογή πελάτη/εξυπηρετητή (client/server) που χρησιμοποιεί ένα φυλλομετρητή (web browser) για το πρόγραμμα πελάτη και «καταναλώνει» μια αλληλεπιδραστική υπηρεσία με την σύνδεση του σε εξυπηρετητές μέσω του διαδικτύου. Μια διαδικτυακή εφαρμογή προβάλλει κατάλληλο δυναμικό περιεχόμενο βασισμένο στις απαιτήσεις του χρήστη, στις καταγεγραμμένες συνήθειες του και σε θέματα ασφάλειας.

Από τον παραπάνω ορισμό είναι ξεκάθαρο πως μια διαδικτυακή εφαρμογή δεν έχει τον στατικό χαρακτήρα προβολής περιεχομένου της απλής ιστοσελίδας αλλά δίνει την δυνατότητα αλληλεπίδρασης και παραγωγής περιεχομένου στον χρήστη. Η αρχιτεκτονική δε που ακολουθεί μια διαδικτυακή εφαρμογή είναι αυτή του πελάτη/εξυπηρετητή. Στο σχήμα που ακολουθεί προβάλλεται η γενική αρχιτεκτονική που ακολουθούν οι διαδικτυακές εφαρμογές.

Σύμφωνα με το παρακάτω σχήμα 5.2 ο φυλλομετρητής του χρήστη επικοινωνεί με το εξυπηρετητή ιστοσελίδας για την προβολή και δημιουργία περιεχομένου. Επίσης ο εξυπηρετητής επικοινωνεί με εξυπηρετητές άλλων σελίδων για την παροχή κατάλληλου περιεχομένου στον χρήστη (πελάτη). Ο πελάτης (χρήστης) μπορεί να αντλεί περιεχόμενο και από άλλους εξυπηρετητές χωρίς να το αιτηθεί σε αυτούς.



*Σχήμα 5.2: Γενική Αρχιτεκτονική Διαδικτυακών εφαρμογών*

### 5.3.2 Τεχνολογίες διαδικτυακού λογισμικού

Σύμφωνα με την αρχιτεκτονική που δόθηκε στην προηγούμενη υποενότητα οι αντίστοιχες τεχνολογίες υλοποίησης διαδικτυακών εφαρμογών πρέπει να υποστηρίζουν τα παρακάτω δομικά στοιχεία της αρχιτεκτονικής:

- Πελάτης (web browser)
- Εξυπηρετητής (web server)
- Επικοινωνία εξυπηρετητών

Για τον πελάτη οι τεχνολογίες υλοποίησης διακρίνονται σε:

- Τεχνολογίες απεικόνισης (HTML, CSS). Τεχνολογίες με τις οποίες το περιεχόμενο προβάλλεται ομοιόμορφα σε διάφορα είδη φυλλομετρητών αλλά και υπολογιστών.
- Προγραμματισμός πελάτη (javascript). Τεχνολογία με την οποία γράφεται κώδικας που εκτελείται από τον φυλλομετρητή του χρήστη. Χρησιμοποιείται για κινούμενα γραφικά, για έλεγχο ορθής εισαγωγής στοιχείων κ.α.

Για τον εξυπηρετητή χρησιμοποιούνται τεχνολογίες προγραμματισμού του (PHP, python, ASP, NODE.js, JSP κ.α.). Ο κώδικας αυτών τεχνολογιών εκτελείται στον εξυπηρετητή και τα αποτελέσματά του μόνο προβάλλονται στον πελάτη. Οι τεχνολογίες αυτές χρησιμοποιούνται για την δημιουργία και προβολή δυναμικού περιεχομένου.

Οι υποδομές ασύγχρονης επικοινωνίας μεταξύ εξυπηρετητών ή πελατών διακρίνονται σε:

- Τεχνολογίες περιγραφής και μεταφοράς δομημένων δεδομένων ( XML)
- Πρότυπο ανταλλαγής δεδομένων (JSON)
- Τεχνική ασύγχρονης επικοινωνίας (AJAX)

Όλες οι παραπάνω τεχνολογίες συνοψίζονται στο παρακάτω σχήμα 5.3.



**Σχήμα 5.3:** Τεχνολογίες διαδικτυακού λογισμικού

### 5.3.3. Τεχνολογίες απεικόνισης

Οι τεχνολογίες απεικόνισης αποτελούν το μέσο με το οποίο η πληροφορία που μεταδίδεται στο διαδίκτυο απεικονίζεται με ενιαίο τρόπο σε οποιοδήποτε υπολογιστικό σύστημα (προσωπικός υπολογιστής PC, tablets, έξυπνα κινητά smartphones) που διαθέτει λογισμικό φυλλομετρητή (browser).

Στην ενότητα αυτή θα μάθουμε να αξιοποιούμε την βασική τεχνολογία απεικόνισης HTML (HyperText Markup Language) για την ανάπτυξη διαδικτυακών εφαρμογών. Επίσης θα μάθουμε να χρησιμοποιούμε την τεχνολογία μορφοποίησης περιεχομένου CSS (Cascade Style Sheet).

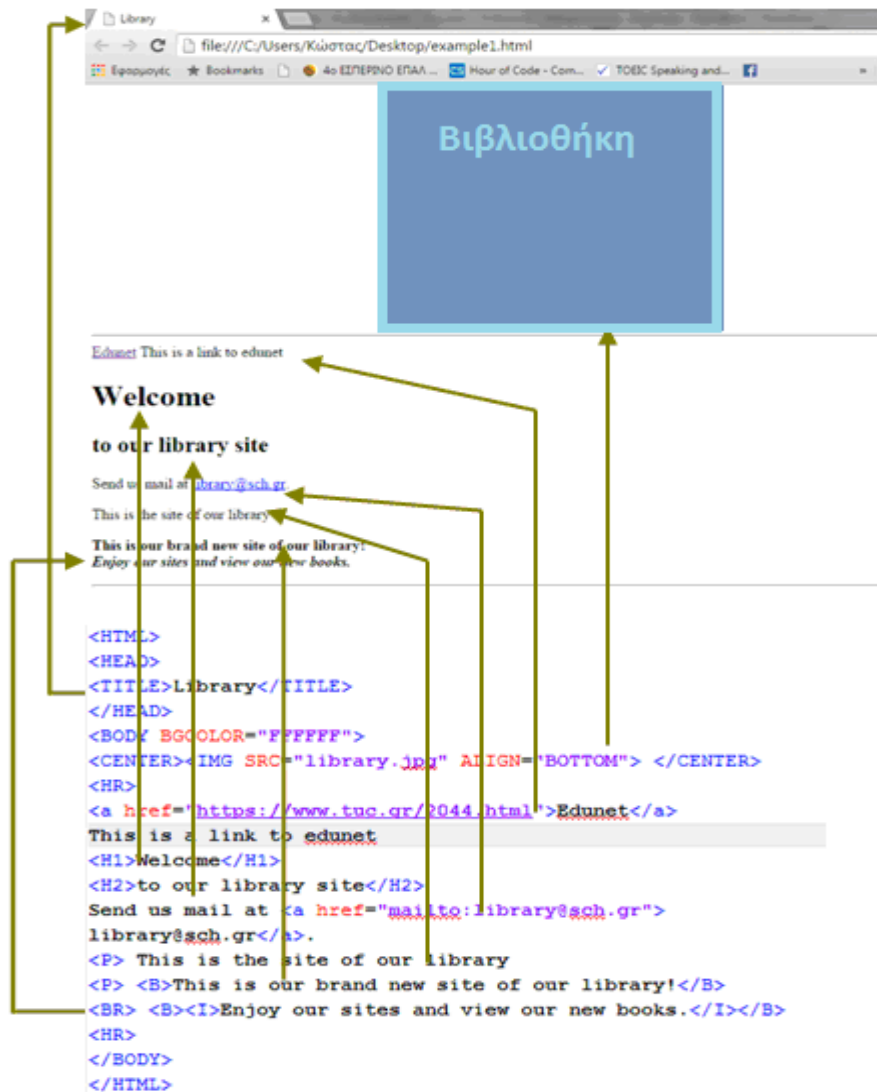
## Εισαγωγή στην HTML και CSS

Στην δεκαετία του 1980, αν και είχαν ήδη δημιουργηθεί οι υποδομές δικτύωσης, προέκυψε το πρόβλημα της συμβατότητας αναπαράστασης της πληροφορίας. Ενώ οι επιστήμονες της εποχής διέθεταν υπολογιστικά συστήματα δικτυωμένα μεταξύ τους η ανταλλαγή πληροφοριών ήταν μια δύσκολη διαδικασία καθώς διέθεταν διαφορετικούς υπολογιστές με διαφορετικά λειτουργικά συστήματα και με διαφορετικά λογισμικά επεξεργασίας και προβολής κειμένου. Η ανάγκη ανταλλαγής

επιστημονικών δεδομένων οδήγησε τον Tim Berners-Lee επιστήμονα του CERN στη ανακάλυψη της γλώσσας απεικόνισης HTML. Σε αυτό το σημείο πρέπει να επισημανθεί ότι η ανάγκη ή καλύτερα η ανακάλυψη ενός κενού (στην καθημερινή ζωή, στο εμπόριο, στις επιστήμες κ.α.) αποτελεί ένα από τα βασικά κίνητρα για την ανακάλυψη και προώθηση της καινοτομίας.

**Συζήτηση στην τάξη:** Με βάση την καθημερινότητα σας, τις γνώσεις σας και τα ενδιαφέροντα σας σκεφτείτε και συζητήστε με τους συμμαθητές σας ιδέες που θα μπορούσαν να αποτελέσουν τεχνολογικές ή/και επιχειρηματικές καινοτομίες. Αφού ακούσετε όλες τις ιδέες των συμμαθητών σας διαλέξτε, χωρίς προσωπικούς ενδοιασμούς τις καλύτερες. Υιοθετήστε τις ιδέες που επιλέχθηκαν και σε ομάδες προτείνετε τρόπους βελτίωσής τους. Είστε έτοιμοι να καινοτομήσετε;

Για την αντιμετώπιση του προβλήματος της συμβατότητας αναπαράστασης της πληροφορίας ο Tim Berners-Lee πρότεινε την δημιουργία μιας γλώσσας περιγραφής της, την HTML. Οποιοδήποτε υπολογιστικό σύστημα μπορεί να αναπαραστήσει αξιόπιστα και ομοιόμορφα την πληροφορία αρκεί να διαθέτει κατάλληλο λογισμικό που διαβάζει και κατανοεί τις “οδηγίες” της HTML και ανάλογα με αυτές εμφανίζει την πληροφορία στην οθόνη. Το λογισμικό αυτό είναι ο γνωστός σε όλους μας ως φυλλομετρητής (browser). Η HTML δεν αποτελεί μια γλώσσα προγραμματισμού όπου οι εντολές που γράφουμε εκτελούνται από τον επεξεργαστή αλλά είναι μια γλώσσα περιγραφής ιστοσελίδας και οι “οδηγίες” της ή όπως θα δούμε στην συνέχεια οι ετικέτες, απευθύνονται στον φυλλομετρητή για να εμφανίσει κατάλληλα το περιεχόμενο. Όλα αυτά απεικονίζονται στο παρακάτω σχήμα 5.4.



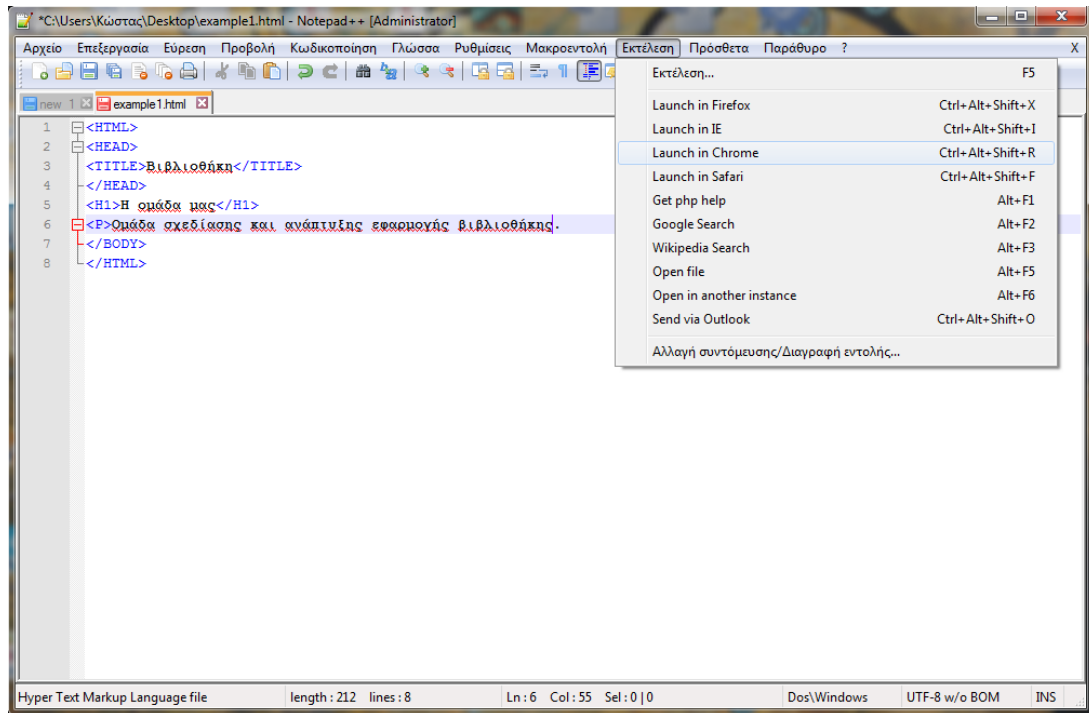
*Σχήμα 5.4: Οδηγίες-ετικέτες HTML στον φυλλομετρητή*

Ενώ, όπως θα δούμε παρακάτω, είναι δυνατή η μορφοποίηση περιεχομένου με την HTML, προέκυψε η ανάγκη να διαχωριστεί το περιεχόμενο από την μορφοποίηση του καθώς οι ιστοσελίδες μεγάλωναν σε όγκο και η συντήρηση-αναβάθμιση τους γινόταν ολοένα και δυσκολότερη. Για αυτό τον λόγο δημιουργήθηκε η τεχνολογία CSS με την οποία μπορεί να γίνει χωριστά η μορφοποίηση των ετικετών της HTML.

### Εργαλεία HTML (notepad++)

Ένα από τα πλεονεκτήματα της HTML είναι ότι χρειάζεται έναν απλό κειμενογράφο (notepad) για την συγγραφή κώδικα και έναν φυλλομετρητή για τον έλεγχο. Βέβαια έχουν δημιουργηθεί πολλά ολοκληρωμένα περιβάλλοντα ανάπτυξης ιστοσελίδων για επαγγελματική χρήση. Στη συνέχεια θα χρησιμοποιήσουμε το λογισμικό ανοικτού κώδικα notepad++ για την συγγραφή κώδικα σε HTML, CSS, Javascript και PHP. Το λογισμικό αυτό προτείνεται διότι είναι “ελαφρύ” στην εκτέλεση του, διευκολύνει την συγγραφή κώδικα με την αυτόματη χρήση χρωμάτων στις δεσμευμένες λέξεις-εντολές της κάθε τεχνολογίας, έχει την δυνατότητα διαχείρισης πολλών εγγράφων και

επιτρέπει την άμεση εκτέλεση του φυλλομετρητή για έλεγχο. Το notepad++ είναι διαθέσιμο για αποθήκευση και εγκατάσταση στην ηλεκτρονική διεύθυνση <https://notepad-plus-plus.org/>. Το περιβάλλον του notepad++ απεικονίζεται στο παρακάτω σχήμα 5.5.



*Σχήμα 5.5: Το λογισμικό notepad++*

Όπως φαίνεται στο σχήμα 5.5 το notepad++ διαθέτει το κλασσικό μενού επιλογών (αρχείο, επεξεργασία, εύρεση, προβολή). Προσοχή πρέπει να δοθεί κατά την αποθήκευση ενός αρχείου όσον αφορά την επέκτασή του. Όλα τα αρχεία HTML αποθηκεύονται με την επέκταση .html για να είναι αναγνώσιμα από τον φυλλομετρητή. Ο κώδικας εισάγεται στην επιφάνεια της κάθε ανοικτής καρτέλας. Χρήσιμη λειτουργία αποτελεί η “εκτέλεση” με την οποία μπορούμε να δούμε την ιστοσελίδα που μόλις δημιουργήσαμε στον επιθυμητό μας φυλλομετρητή.

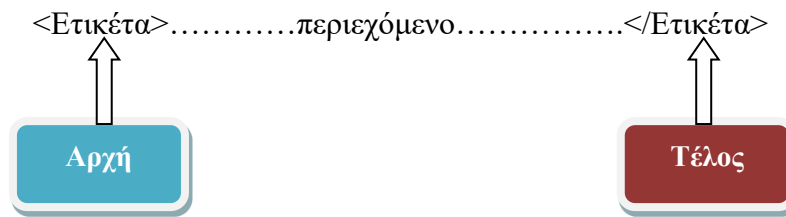
**Δραστηριότητα:** Ανοίξτε το notepad++ και δημιουργήστε ένα νέο αρχείο με επέκταση .html (myfisrtpage.html). Στην συνέχεια εισάγετε τον κώδικα που βλέπετε στο παραπάνω σχήμα 5.5. Εκτελέστε τον κώδικα στον διαθέσιμο φυλλομετρητή. Επισκεφτείτε την ηλεκτρονική διεύθυνση

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_basic](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_basic)

εισάγετε τον ίδιο κώδικα στο κατάλληλο πλαίσιο και δείτε το αποτέλεσμα στο διπλανό πλαίσιο. Αναζητήστε στο διαδίκτυο και άλλους online HTML editors. Συγκρίνετε τους offline με τους online HTML editors. Είναι δικαιολογημένη η τάση ανάπτυξης ακόμα και των κλασσικών απλών εφαρμογών στο διαδίκτυο;

## Βασικές ετικέτες στην HTML

Οι περισσότερες ετικέτες στην HTML ακολουθούν την παρακάτω γενική δομή



*Σχήμα 5.6: Γενική δομή ετικέτας HTML*

Η αρχή κάθε ετικέτας επισημαίνεται με το όνομα της ετικέτας ανάμεσα στα δύο < σύμβολα ενώ το τέλος επισημαίνεται με το σύμβολο / και το όνομα της ετικέτας ανάμεσα στα δύο σύμβολα < > όπως φαίνεται στο παραπάνω σχήμα 5.6. Ανάμεσα στη αρχή και το τέλος παρεμβάλλεται το περιεχόμενο που απεικονίζεται με αυτήν την ετικέτα. Το τέλος της ετικέτας δεν είναι απαραίτητο να βρίσκεται στην ίδια γραμμή με την αρχή και μπορεί να γραφεί σε επόμενη γραμμή. Επιπροσθέτως, μια ετικέτα μπορεί να εμπεριέχει ως περιεχόμενο και άλλες ετικέτες.

Η βασική δομή που ακολουθεί οποιοδήποτε αρχείο HTML φαίνεται στο παρακάτω σχήμα 5.7.

```
<HTML>
  <HEAD>
    <TITLE> ΤΙΤΛΟΣ ΣΕΛΙΔΑΣ </TITLE>
  </HEAD>
  <BODY>
    .....
    ΚΥΡΙΟ ΠΕΡΙΕΧΟΜΕΝΟ
  </BODY>
</HTML>
```

*Σχήμα 5.7: Βασική δομή αρχείου HTML*

Η ετικέτα <HTML> προσδιορίζει την αρχή και το τέλος ενός αρχείου HTML. Ένα HTML αρχείο αποτελείται από δύο μέρη. Την επικεφαλίδα που προσδιορίζεται από την ετικέτα <HEAD> και το κυρίως σώμα που προσδιορίζεται από την ετικέτα <BODY>. Στην επικεφαλίδα μπορεί να μπει η ετικέτα <TITLE>, ο τίτλος της σελίδας όπως διαφαίνεται στην καρτέλα της ιστοσελίδας στον φυλλομετρητή, ετικέτες σχετικές με μορφοποίηση καθώς και ετικέτες-λέξεις κλειδιά που θα δούμε αναλυτικά σε επόμενη υποενότητα. Στο κύριο μέρος μπαίνει όλο το περιεχόμενο της ιστοσελίδας με οποιαδήποτε μορφή.

Για την παρουσίαση περιεχομένου σε παράγραφο χρησιμοποιείται η ετικέτα <p>. Το τέλος σε αυτήν την ετικέτα δεν είναι απαραίτητο. Για την δημιουργία κειμένου κεφαλίδας χρησιμοποιούνται οι ετικέτες <H1>, <H2>, <H3>, <H4>, <H5> και <H6>. Για την εισαγωγή εικόνας χρησιμοποιείται η ετικέτα <img>, όπως φαίνεται στην παρακάτω δήλωση.

```

```

Η παράμετρος src της ετικέτας <img> αποτελεί το όνομα του αρχείου της εικόνας που θα προβληθεί. Το αρχείο εικόνας πρέπει να βρίσκεται στο ίδιο φάκελο με το αρχείο HTML διαφορετικά στο src πρέπει να γραφεί όλη η διαδρομή θέσης του αρχείου εικόνας. Οι τύποι αρχείων εικόνας που υποστηρίζονται από την HTML είναι οι: JPEG, GIF και PNG. Στην παράμετρο style ορίζεται ένα παράθυρο σε pixels με τις ιδιότητες width (πλάτος) και height (ύψος) μέσα στο οποίο θα προσαρμοστεί η εικόνα.

Για την δημιουργία συνδέσμων (links - λέξεις ή εικόνες με τις οποίες ο χρήστης μπορεί να μεταβεί σε άλλο κόμβο πληροφορίας) χρησιμοποιείται η ετικέτα <a> , όπως φαίνεται στην παρακάτω δήλωση.

```
<a href="http://www.sch.gr">Σχολικό Δίκτυο</a>
```

Σύμφωνα με το παραπάνω παράδειγμα οι λέξεις «Σχολικό δίκτυο» αποτελούν σύνδεσμο. Κάθε φορά που ο χρήστης κάνει click σε αυτές τις λέξεις ο φυλλομετρητής φορτώνει το περιεχόμενο της σελίδας που βρίσκεται στην παράμετρο href της ετικέτας <a>. Στο συγκεκριμένο παράδειγμα θα φορτώσει την σελίδα [www.sch.gr](http://www.sch.gr). Η λειτουργία όλων αυτών των ετικετών φαίνονται στην πράξη στο παρακάτω σχήμα 5.8.

```
<HTML>
<HEAD>
<TITLE>Βιβλιοθήκη</TITLE>
</HEAD>
<BODY>

<H1>Βιβλιοθήκη μας</H1>
<H2>Βιβλιοθήκη μας</H2>
<H3>Βιβλιοθήκη μας</H3>
<H4>Βιβλιοθήκη μας</H4>
<H5>Βιβλιοθήκη μας</H5>
<H6>Βιβλιοθήκη μας</H6>
<P>Καλώς ήρθατε στην ιστοσελίδα του σχολείου μας</P>
<P>Εδώ μπορείτε να ενημερωθείτε για τα βιβλία που μπορείτε να δανειστείτε</P>
</BODY>
</HTML>
```

*Σχήμα 5.8: Παράδειγμα βασικών HTML ετικετών*

**Δραστηριότητα:** Εισάγετε τον κώδικα του παραπάνω παραδείγματος στο notepad++ και κάντε τους απαραίτητους ελέγχους. Εμφανίζεται η εικόνα; Διορθώστε το



πρόβλημα. Δημιουργήστε μια νέα σελίδα όπου ως τίτλο θα έχει το μικρό σας όνομα, θα εμφανίζεται η φωτογραφία σας (selfie) καθώς και η φράση «Μερικά λόγια για μένα» με κεφαλίδα της αρεσκείας σας και τέλος θα περιέχει μια σύντομη περιγραφή του εαυτού σας.

## Λίστες και πίνακες στην HTML

Στην προηγούμενη ενότητα μάθαμε τις βασικές ετικέτες της HTML με τις οποίες μπορούμε να αποδώσουμε πληροφορία με την μορφή κειμένου και/ή εικόνας. Πολλές φορές όμως δεν αρκεί μόνο η παρουσίαση της πληροφορίας αλλά πρέπει να αποδίδεται το πώς αυτή είναι δομημένη. Στην υποενότητα αυτή θα μάθουμε τις βασικές ετικέτες δόμησης πληροφορίας, τις λίστες και τους πίνακες.

Οι λίστες που είναι διαθέσιμες στην HTML είναι οι **αριθμημένες (ordered)** και οι **μη αριθμημένες (unordered)**. Οι αριθμημένες λίστες εμφανίζουν δεδομένα με μια αύξουσα σειρά αρίθμησης ενώ οι μη αριθμημένες λίστες δεν εμφανίζουν σειρά αρίθμησης. Η ετικέτα για την δημιουργία αριθμημένης λίστας είναι η <OL> και η ετικέτα που χρησιμοποιείται για την δημιουργία αντικειμένου μέσα στην λίστα είναι η <LI>. Για τη δημιουργία μη αριθμημένης λίστας χρησιμοποιείται η ετικέτα <UL> ενώ για την δημιουργία αντικειμένων μέσα στην λίστα χρησιμοποιείται πάλι η ετικέτα <LI>. Όλα αυτά φαίνονται στο παρακάτω σχήμα 5.9.

```
<HTML>
  <HEAD>
    <TITLE>Βιβλιοθήκη</TITLE>
  </HEAD>
  <BODY>
    <H1>Κατηγορίες Βιβλίων</H1>
    <UL>
      <LI>Πληροφορικής</LI>
      <LI>Μαθηματικά</LI>
      <LI>Ιστορία</LI>
      <LI>Μυθιστορήματα</LI>
    </UL>

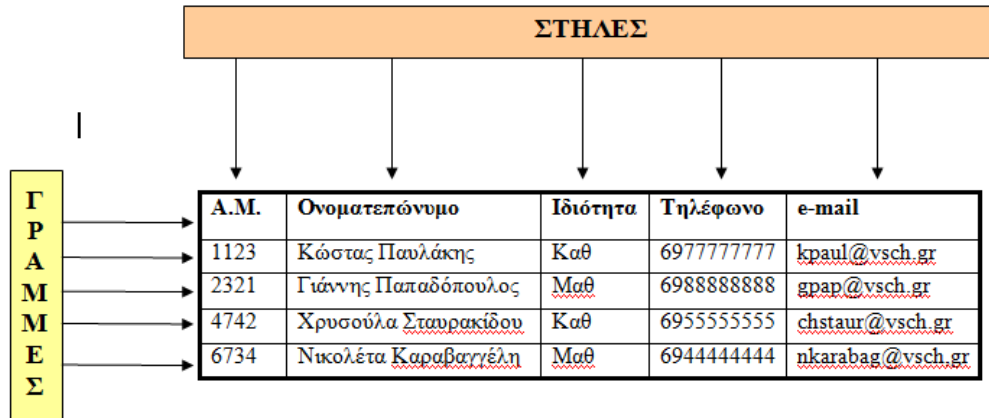
    <H1>Μαθητές που έχουν διαβάσει τα περισσότερα βιβλία</H1>
    <OL>
      <LI>Κώστας Χ.</LI>
      <LI>Μαρία Κ.</LI>
      <LI>Ιουλιάννα Α.</LI>
      <LI>Κατερίνα Ε.</LI>
    </OL>

  </BODY>
</HTML>
```

Σχήμα 5.9: Λίστες στην HTML

**Δραστηριότητα:** Εισάγετε τον κώδικα του παραπάνω παραδείγματος στο notepad++ και κάντε τους απαραίτητους ελέγχους. Ποια είναι η διαφορά των δύο ειδών λιστών;

Η πιο κλασσική δομή δεδομένων είναι αυτή του πίνακα. Σε έναν πίνακα τα δεδομένα οργανώνονται σε γραμμές και στήλες όπως φαίνεται στο παρακάτω σχήμα 5.10.



*Σχήμα 5.10: Δομή δεδομένων – Πίνακας*

Η ετικέτα <TABLE> χρησιμοποιείται για την δημιουργία πίνακα στην HTML. Η ετικέτα <TR> χρησιμοποιείται για την δημιουργία γραμμών μέσα στον πίνακα ενώ η ετικέτα <TD> χρησιμοποιείται για την δημιουργία κελιών μέσα στην γραμμή (αν ο αριθμός κελιών είναι ο ίδιος σε κάθε γραμμή τότε αυτός ο αριθμός είναι ο αριθμός των στηλών). Η χρήση αυτών των ετικετών φαίνεται στο παρακάτω σχήμα 5.11.

```
<HTML>
<HEAD>
  <TITLE>Βιβλιοθήκη</TITLE>
</HEAD>
<BODY>
  <H1>Κατάλογος Βιβλίων</H1>
  <TABLE border="1px">
    <TR>
      <TD>ISBN</TD>
      <TD>TITLE</TD>
      <TD>YEAR PUBLISHED</TD>
    </TR>
    <TR>
      <TD>0778983932</TD>
      <TD>Introduction to JAVA</TD>
      <TD>2009</TD>
    </TR>
    <TR>
      <TD>0028892822</TD>
      <TD>Arduino projects</TD>
      <TD>2011</TD>
    </TR>
  </TABLE>
</BODY>
</HTML>
```

*Σχήμα 5.11: Πίνακας στην HTML*

**Δραστηριότητα:** Εισάγετε τον κώδικα του παραπάνω παραδείγματος στο notepad++ και κάντε τους απαραίτητους ελέγχους. Αναζητήστε στο διαδίκτυο το ρόλο της ιδιότητας border του πίνακα. Προσθέστε μια νέα στήλη με τίτλο θεματολογία. Για κάθε βιβλίο θα υπάρχει συνοπτική θεματολογία με την μορφή μη αριθμημένης λίστας.

## Φόρμες στην HTML

Στις προηγούμενες ενότητες μάθαμε ετικέτες με τις οποίες απεικονίζουμε πληροφορία. Παρόλα αυτά τα σύγχρονα online πληροφοριακά συστήματα δίνουν την δυνατότητα αλληλεπίδρασης στον χρήστη, δηλαδή την δυνατότητα εισαγωγής δεδομένων από τον χρήστη στο πληροφοριακό σύστημα. Η ετικέτα <FORM> της HTML επιτρέπει την εισαγωγή γραφικών στοιχείων με τα οποία ο χρήστης μπορεί να εισάγει πληροφορία. Σε αυτό το σημείο πρέπει να διευκρινιστεί ότι η HTML δεν αρκεί για την αποστολή και αποθήκευση δεδομένων χρήστη αλλά χρειάζεται και προγραμματισμός στον εξυπηρετητή (PHP), γνώσεις που θα περιγραφούν σε επόμενη ενότητα. Η χρήση της ετικέτας <FORM> περιγράφεται στο σχήμα 5.12.

```
<HTML>
<HEAD>
  <TITLE>Βιβλιοθήκη</TITLE>
</HEAD>
<BODY>
  <H1>Εγγραφή νέου χρήστη</H1>

  <form action="action_page.php">
    First name:<br>
    <input type="text" name="firstname" value="">
    <br>
    Last name:<br>
    <input type="text" name="lastname" value="">
    <br>
    e-mail:<br>
    <input type="text" name="email" value="">
    <br>
    phone:<br>
    <input type="text" name="phone" value="">
    <br>
    password:<br>
    <input type="text" name="password" value="">
    <br>
    retype password:<br>
    <input type="text" name="repassword" value="">
    <br><br>
    <input type="submit" value="Submit">
  </form>
</BODY>
</HTML>
```

*Σχήμα 5.12: Η ετικέτα <FORM>*

Βασική ιδιότητα της ετικέτας <FORM> είναι η action, με την οποία δηλώνεται το αρχείο PHP του οποίου ο κώδικας θα εκτελεστεί στον εξυπηρετητή όταν πατηθεί το κουμπί της φόρμας. Η ετικέτα <input> χρησιμοποιείται για την εισαγωγή γραφικών στοιχείων στη φόρμα. Η ιδιότητα type υποδηλώνει τον τύπο του γραφικού στοιχείου, δηλαδή αν είναι πλαίσιο κειμένου (text), κουμπί ενέργειας (submit) και άλλα. Η ιδιότητα name αποτελεί το αναγνωριστικό όνομα του γραφικού στοιχείου για την περεταίρω επεξεργασία των δεδομένων του από την PHP ενώ η ιδιότητα value περιέχει τα δεδομένα του γραφικού στοιχείου.

**Δραστηριότητα:** Εισάγετε τον κώδικα του παραπάνω παραδείγματος στο notepad++ και κάντε τους απαραίτητους ελέγχους. Αναζητήστε στο διαδίκτυο το γραφικό στοιχείο radiobutton και χρησιμοποιώντας το δώστε την επιλογή στο χρήστη να επιλέγει το φύλο.

### 5.3.4 Τεχνολογίες προγραμματισμού πελάτη (client side)

Στην προηγούμενη ενότητα μάθαμε για την HTML, τεχνολογία με την οποία μπορούμε να απεικονίσουμε την πληροφορία στατικά. Δεν μπορούμε όμως να απεικονίσουμε την δυναμική της πληροφορίας. Για αυτό το λόγο χρησιμοποιούνται οι τεχνολογίες προγραμματισμού πελάτη (client side) με τις οποίες μπορούμε να εισάγουμε κώδικα που εκτελείται δυναμικά και αλληλεπιδρά με τα στοιχεία της HTML. Ο κώδικας αυτός εκτελείται στον φυλλομετρητή του χρήστη (client). Διάφορες τεχνολογίες έχουν δημιουργηθεί για αυτό το σκοπό όπως είναι η vbscript και η Javascript . Στις επόμενες ενότητες θα μάθουμε τις βασικές έννοιες της Javascript έτσι ώστε να μπορούμε να την χρησιμοποιούμε αλλά και να αποκτήσουμε τις βάσεις για περεταίρω εκμάθηση αυτής της τεχνολογίας.

#### Εισαγωγή στην Javascript

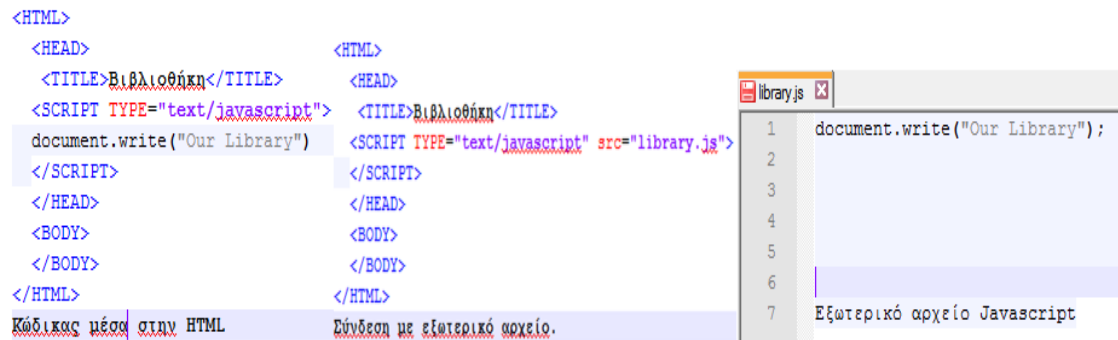
Η Javascript αποτελεί μια από τις πιο διαδεδομένες γλώσσες συγγραφής σεναρίων (scripting language) για προγραμματισμό στην πλευρά του πελάτη (client side). Έχει πολλές δυνατότητες και χρησιμοποιείται για τον εμπλουτισμό ενός online πληροφοριακού συστήματος με αυξημένη λειτουργικότητα και αλληλεπίδραση. Το σημαντικότερο πλεονέκτημά της είναι ότι εκτελείται στο πελάτη (στον φυλλομετρητή) καθιστώντας την εφαρμογή ολοένα και πιο γρήγορη. Επίσης η Javascript χρησιμοποιείται από πολλούς προγραμματιστές για αυτό και υπάρχουν άφθονοι πόροι στο διαδίκτυο τόσο για την εκμάθησή της όσο και για την γρήγορη επίλυση προβλημάτων (έτοιμες και πολλές βιβλιοθήκες είναι ελεύθερες προς χρήση).

Η Javascript ενσωματώνεται σε ένα αρχείο HTML με την ετικέτα <SCRIPT> της HTML. Ενδιάμεσα σε αυτή την ετικέτα μπορεί να γραφεί κώδικας σε Javascript. Επίσης κώδικας σε αυτή τη γλώσσα μπορεί να γραφεί σε εξωτερικό αρχείο με κατάληξη .js (π.χ. library.js). Η σύνδεση με το αρχείο αυτό γίνεται μέσω της ιδιότητας src της ετικέτας <SCRIPT>. Η ετικέτα <SCRIPT> μπορεί να ενσωματωθεί

σε οποιοδήποτε μέρος μιας σελίδας HTML (BODY ή HEAD). Η πρώτη εντολή της Javascript (εμφάνιση μηνύματος “our library” στην οθόνη) που θα μάθουμε είναι η:

**document.write(“Our Library”);**

Όλα τα παραπάνω συνοψίζονται στο σχήμα 5.13.



**Σχήμα 5.13:** Εισαγωγή στην Javascript

Μπορούμε να χρησιμοποιήσουμε το Notepad++ για την δημιουργία εξωτερικών αρχείων τύπου .js.

**Δραστηριότητα:** Εισάγετε τον κώδικα των παραπάνω δύο παραδειγμάτων στο notepad++ και κάντε τους απαραίτητους ελέγχους.

### Μεταβλητές και πίνακες στην Javascript

Όπως όλες οι γλώσσες προγραμματισμού έτσι και η Javascript υποστηρίζει τον μηχανισμό προσωρινής αποθήκευσης δεδομένων στην μνήμη RAM, δηλαδή τον μηχανισμό των μεταβλητών. Οι τύποι δεδομένων που υποστηρίζονται από αυτήν είναι:

- Αριθμοί (ακέραιοι – δεκαδικοί)
- Αλφαριθμητικά (strings)
- Λογικές ( true–false)
- null (κενή μεταβλητή)
- undefined (μη προσδιορισμένη)

Οι μεταβλητές στην Javascript δημιουργούνται με την δεσμευμένη λέξη var, το όνομα της μεταβλητής, το σύμβολο εκχώρησης = και την τιμή που θέλουμε να εκχωρήσουμε. Ο τύπος δεδομένων της μεταβλητής ορίζεται αυτόματα από τον τύπο δεδομένων της εκχωρούμενης τιμής. Όλα αυτά συνοψίζονται στα παρακάτω παραδείγματα δημιουργίας μεταβλητών.

- var number = 8;
- var pi = 3.14;

- `var book_title = "Internet programming ";`
- `var book_year = "2009";`
- `var booked = FALSE;`
- `var temp = null;`
- `var feature = undefined;`

Οι βασικοί τελεστές που χρησιμοποιούνται για τη επεξεργασία μεταβλητών είναι οι: +, -, \*, / και η χρήση τους φαίνεται στα παρακάτω παραδείγματα (τα σύμβολα // προσθέτουν σχόλια στον κώδικα).

- `number = number + 8; // αυξάνεται η τιμή της number κατά 8`
- `number += 2; // αυξάνεται η τιμή της number κατά 2`
- `book_title += book_year // Η μεταβλητή book_title θα αποκτήσει την τιμή "Internet programming 2009"`
- `number = number - 2 // Η τιμή του number μειώνεται κατά 2`
- `number = number * 3 // Η τιμή του number πολλαπλασιάζεται με το 3`
- `number = number / 4 // Η τιμή του number διαιρείται με το 4`

Η Javascript επιτρέπει την δημιουργία πινάκων, δηλαδή την προσωρινή αποθήκευση στην μνήμη RAM πολλαπλών δεδομένων με το ίδιο όνομα. Μπορούμε να ξεχωρίσουμε τα δεδομένα με την χρήση δείκτη. Στην Javascript δεν είναι απαραίτητο όλα τα στοιχεία ενός πίνακα να είναι του ίδιου τύπου. Συγκεκριμένα ένας πίνακας στην Javascript δημιουργείται με τους τρεις εναλλακτικούς τρόπους όπως φαίνεται παρακάτω:

**A) `var Books = new Array();`**

**B) `var Books = new Array("Learning JAVA", "Data structures in C++", "Arduino projects");`**

**Γ) `var Books = ["Learning JAVA", "Data structures in C++", "Arduino projects"];`**

Με τον A) τρόπο δημιουργείται ο κενός πίνακας Books. Οι τρόποι B) και Γ) είναι ισοδύναμοι και δημιουργούν τον πίνακα Books τριών θέσεων όπου σε κάθε θέση έχει τον τίτλο του κατά σειρά βιβλίου. Η αρίθμηση των θέσεων του πίνακα ξεκινάει από το 0. Έτσι το πρώτο στοιχείο είναι το Books[0], το δεύτερο το Books[1] και το τρίτο το Books[2]. Χρησιμοποιώντας την εντολή:

**`document.write(Books[1]);`**

θα τυπωθούν στην ιστοσελίδα τα περιεχόμενα της δεύτερης θέσης του πίνακα Books, δηλαδή το αλφαριθμητικό "Data structures in C++". Ενώ με την εντολή εκχώρησης:

**`Books[2] = "Raspberry PI in action";`**

τα περιεχόμενα της τρίτης θέσης του πίνακα Books θα αλλάξουν σε “Raspberry PI in action”. Μια βασική ιδιότητα των πινάκων είναι η length (πλήθος στοιχείων πίνακα). Έτσι το Books.length μας δίνει 3 όσο και το πλήθος των στοιχείων. Αν θέλουμε να διαγράψουμε τα δυο τελευταία στοιχεία του πίνακα θα γράψουμε:

**Books.length = 1;**

Η Javascript διαθέτει αρκετές μεθόδους για την επεξεργασία πινάκων (λειτουργίες πάνω στους πίνακες). Για παράδειγμα η μέθοδος reverse() αντιστρέφει την σειρά των στοιχείων. Έτσι η εντολή:

**Books.reverse();**

Βάζει στο πρώτο στοιχείο την τιμή “Arduino projects” ενώ στο τρίτο βάζει την τιμή “Learning JAVA”. Μια άλλη μέθοδος είναι η push() που εισάγει στο τέλος του πίνακα ένα νέο στοιχείο.

**Books.push(“Internet of Things”);**

**Συζήτηση στην τάξη:** Ποια θα είναι η μορφή του πίνακα Books μετά την εκτέλεση της παραπάνω εντολής; Υπάρχουν άλλες μέθοδοι – ιδιότητες για έναν πίνακα; Αναζητήστε στο διαδίκτυο.

### Δομές επιλογής-επανάληψης στην Javascript.

Όπως σε όλες τις γλώσσες προγραμματισμού έτσι και στην Javascript υπάρχουν οι δομές επιλογής και επανάληψης. Πριν προχωρήσουμε στην περιγραφή αυτών των δομών θα περιγραφούν οι τελεστές σύγκρισης και οι λογικοί τελεστές της γλώσσας. Οι τελεστές σύγκρισης μαζί με την λειτουργία τους δίνονται παρακάτω:

- < , μικρότερο
- > , μεγαλύτερο
- <= , μικρότερο ή ίσο
- >= , μεγαλύτερο ή ίσο
- == , ίσο (προσοχή να μην συγχέεται με τον τελεστή εκχώρησης =)
- != , όχι ίσο
- === , αυστηρά ίσο (ίσο σε τιμή και σε τύπο δεδομένων)
- !== , αυστηρά μη ίσο (μη ίσο σε τιμή και σε τύπο δεδομένων)

Ακολουθούν οι λογικοί τελεστές σύγκρισης με τους οποίους μπορεί να δημιουργηθούν σύνθετες λογικές συνθήκες:

- && , λογικός τελεστής ΚΑΙ
- || , λογικός τελεστής Η’
- ! , λογικός τελεστής ΟΧΙ (αντιστρέφει μια λογική πρόταση)

Στην συνέχεια θα περιγραφεί η βασική δομή επιλογής **if**, σε δύο διαφορετικές εκδοχές με την μορφή παραδειγμάτων. Εξυπακούεται πως μπορούν χρησιμοποιηθούν και πιο σύνθετες συνθήκες.

```
var limit = 10;
var num_of_books = 5;
if (num_of_books < limit){
document.write("Μπορείτε να δανειστείτε")
}
```

*Σχήμα 5.14: Η δομή επιλογής if*

Σε αυτή την εκδοχή η δομή if ελέγχει εάν ισχύει η συνθήκη μέσα στις παρενθέσεις (). Σε περίπτωση που ισχύει εκτελείται ο κώδικας ανάμεσα στις αγκύλες {}. Στην αντίθετη περίπτωση δεν εκτελείται ο κώδικας ανάμεσα στις {} και η ροή του προγράμματος συνεχίζει μετά την δομή if. Στο παράδειγμα αυτό ισχύει η συνθήκη. Κατά συνέπεια θα εκτελεστεί ο «ελεγχόμενος» κώδικας και θα τυπωθεί στην ιστοσελίδα το μήνυμα “Μπορείτε να δανειστείτε”.

```
var limit = 10;
var num_of_books = 10;
if (num_of_books < limit){
document.write("Μπορείτε να δανειστείτε");
}else {
document.write("Δεν μπορείτε να δανειστείτε");
}
```

*Σχήμα 5.15: Η δομή επιλογής if-else*

Σε αυτή την εκδοχή η δομή if ελέγχει εάν ισχύει η συνθήκη μέσα στις παρενθέσεις. Σε περίπτωση που ισχύει εκτελείται μόνο ο κώδικας ανάμεσα στις πρώτες αγκύλες {}. Στην αντίθετη περίπτωση εκτελείται μόνο ο κώδικας ανάμεσα στις δεύτερες αγκύλες else {}. Η ροή του προγράμματος συνεχίζει μετά την δομή if. Στο παράδειγμα αυτό δεν ισχύει η συνθήκη οπότε θα εκτελεστεί ο «ελεγχόμενος» κώδικας του else και θα τυπωθεί στην ιστοσελίδα το μήνυμα “Δεν μπορείτε να δανειστείτε”.

Η Javascript διαθέτει δύο δομές επανάληψης την **while** και την **for**. Η **while** χρησιμοποιείται όταν, κατά την σχεδίαση-συγγραφή του προγράμματος, δεν είναι γνωστός ο αριθμός των επαναλήψεων ενώ η **for** χρησιμοποιείται όταν αυτός ο αριθμός είναι γνωστός. Ακολουθούν οι επεξηγήσεις των δομών αυτών μέσω παραδειγμάτων. Εξυπακούεται πως μπορούν χρησιμοποιηθούν και πιο σύνθετες συνθήκες στην περίπτωση της **while**.



```

<HTML>
  <HEAD>
    <TITLE>Βιβλιοθήκη</TITLE>
    <SCRIPT TYPE="text/javascript">
      var Books = new Array("Absolute JAVA", "Learning PHP", "Hardware design");
      var i = 0;

      while (i < Books.length) {
        document.write(Books[i]);
        document.write("<br />");
        i++;
      }
    </SCRIPT>
  </HEAD>
  <BODY>
  </BODY>
</HTML>

```

*Σχήμα 5.16: Η δομή επανάληψης while*

Στην δομή επανάληψης while ο κώδικας που βρίσκεται ανάμεσα στις αγκύλες {} επαναλαμβάνεται εφόσον ισχύει η συνθήκη στις παρενθέσεις (). **Προσοχή!** Στον επαναλαμβανόμενο κώδικα πρέπει να υπάρχουν εντολές που κάποια στιγμή θα αλλάζουν τις τιμές μεταβλητής/των έτσι ώστε κάποια στιγμή να μην ισχύει η συνθήκη και η ροή του προγράμματος να μεταφερθεί κάτω από την δομή while. Σε αυτό το παράδειγμα ελέγχεται αν η μεταβλητή i είναι μικρότερη από το μήκος του πίνακα, εάν είναι, τυπώνεται το στοιχείο του πίνακα Books που βρίσκεται στην θέση i. Σε κάθε επανάληψη η i αυξάνεται κατά 1(i++). Με αυτόν τον τρόπο τυπώνονται όλα τα στοιχεία του πίνακα Books (εφόσον η αρχική τιμή του i είναι 0, αρχικοποίηση πριν την δομή while).

```

<HTML>
  <HEAD>
    <TITLE>Βιβλιοθήκη</TITLE>
    <SCRIPT TYPE="text/javascript">
      var Books = new Array("Absolute JAVA", "Learning PHP", "Hardware design");

      for (var i = 0; i < Books.length; i++) {
        document.write(Books[i]);
        document.write("<br />");
      }
    </SCRIPT>
  </HEAD>
  <BODY>
  </BODY>
</HTML>

```

*Σχήμα 5.17: Η δομή επανάληψης for*

Στην δομή επανάληψης for χρησιμοποιείται μια μεταβλητή που έχει τον ρόλο του μετρητή των επαναλήψεων. Το τμήμα ελέγχου της δομής επανάληψης, που βρίσκεται ανάμεσα στις παρενθέσεις, αποτελείται από τρία τμήματα α) Αρχικοποίηση της μεταβλητής-μετρητή (δίνεται η αρχική τιμή της μεταβλητής) var i = 0; β) Συνθήκη

ελέγχου μεταβλητής-μετρητή (λογική συνθήκη,  $i < \text{Books.length}$ , που περιέχει οπωσδήποτε την μεταβλητή-μετρητή, όσο ισχύει η συνθήκη επαναλαμβάνεται η εκτέλεση των εντολών που βρίσκονται ανάμεσα στις αγκύλες `{}`) γ) ρυθμός μεταβολής μεταβλητής-μετρητή σε κάθε επανάληψη `i++` (αύξηση κατά 1, στο τμήμα αυτό δηλώνεται το πόσο θα αυξάνεται η μεταβλητή-μετρητής σε κάθε επανάληψη). Το παράδειγμα στο σχήμα 5.17 (for) έχει ακριβώς το ίδιο αποτέλεσμα με το παράδειγμα στο σχήμα 5.16 (while).

**Δραστηριότητα:** Εισάγετε τον κώδικα των παραπάνω παραδειγμάτων στο notepad++ και κάντε τους απαραίτητους ελέγχους. Δημιουργήστε πρόγραμμα σε Javascript που θα εκχωρεί σε έναν πίνακα 10 τίτλους βιβλίων ενώ σε ένα άλλο πίνακα θα αποθηκεύει το αν το κάθε βιβλίο είναι κρατημένο ή όχι (true/false). Κατόπιν το πρόγραμμα θα τυπώνει τα βιβλία που είναι ελεύθερα και το σύνολο αυτών.

### Αλληλεπίδραση Javascript με στοιχεία HTML (DOM)

Η Javascript αλληλεπιδρά με ετικέτες της HTML μέσω του μοντέλου DOM (Document Object Model). Το DOM αποτελεί μια ιδεατή δομή της ιστοσελίδας έτσι ώστε η Javascript να έχει πρόσβαση στα στοιχεία της. Η βασική εντολή της DOM (με τις παραλλαγές της) είναι η:

**`document.getElementById("someId").value;`**

**`document.getElementById("someId").name;`**

**`document.getElementById("someId").innerHTML;`**

Η εντολή `document.getElementById("someId")` επιστρέφει το στοιχείο της HTML που έχει την ιδιότητα `id` ίση με `"someId"`. Γι αυτό το λόγο στα στοιχεία της HTML που θέλουμε να αλληλεπιδρούν με την Javascript ορίζουμε την ιδιότητα `id`. Προσθέτοντας την ιδιότητα `.value` η εντολή επιστρέφει την τιμή της ιδιότητας `value` του στοιχείου HTML. Ομοίως συμβαίνει προσθέτοντας την ιδιότητα `.name`. Ενώ προσθέτοντας την ιδιότητα `innerHTML` η εντολή επιστρέφει το περιεχόμενο του στοιχείου. Αυτά απεικονίζονται στο σχήμα 5.18.

```

<HTML>
  <HEAD>
    <TITLE>Βιβλιοθήκη</TITLE>

  </HEAD>
  <BODY>
    <H1 id="et1"> Our library site</h1>
    <H2 id="et2">Find amazing books</H2>
    <p id="par1"> find the books you want</p>

    <SCRIPT TYPE="text/javascript">
      var s1 = document.getElementById("et1").innerHTML;
      document.write(s1);
      document.write("<br />");
      var s2 = document.getElementById("et2").innerHTML;
      document.write(s2);
      document.write("<br />");
      var s3 = document.getElementById("par1").innerHTML;
      document.write(s3);
      document.write("<br />");
      document.getElementById("par1").innerHTML = "Ran out of books"
    </SCRIPT>

  </BODY>
</HTML>

```

*Σχήμα 5.18: Η εντολή document.getElementById*

Στο παράδειγμα του σχήματος 5.18 η Javascript, μέσω της παραπάνω εντολής του DOM, διαβάζει τα περιεχόμενα των ετικετών <H1>, <H2> και <p> και τα ξανατυπώνει στην ιστοσελίδα. Η τελευταία εντολή του script κάνει ακριβώς το αντίθετο. Εκχωρεί μέσω του DOM μια νέα τιμή “Ran out of books” και το περιεχόμενο της ετικέτας <p> αλλάζει δυναμικά.

Μια άλλη εντολή του DOM που χρησιμοποιείται για διασύνδεση στοιχείων φόρμας <FORM> είναι η:

**document.form[0].element[0].value/name/innerHTML;**

Το form[0] αναφέρεται στην πρώτη φόρμα της ιστοσελίδας, το form[1] στην δεύτερη, το form[2] στην τρίτη και γενικότερα το form[n] στην n+1 φόρμα για την περίπτωση που έχουμε παραπάνω από μια φόρμες σε μια ιστοσελίδα. Συνήθως έχουμε μια φόρμα οπότε το form[0] αρκεί. Το element[0] αναφέρεται στο πρώτο στοιχείο της φόρμας (π.χ. εάν αυτό είναι πλαίσιο κειμένου, τότε θα αναφέρεται σε αυτό), το element[1] στο δεύτερο στοιχείο της φόρμας, το element[2] στο τρίτο στοιχείο της φόρμας και γενικότερα το element[n] στο n+1 στοιχείο της φόρμας. Οι ιδιότητες value/name/innerHTML έχουν τις ίδιες λειτουργίες όπως περιγράφηκαν στο παράδειγμα της getElementById.

**Δραστηριότητα:** Εισάγετε τον κώδικα των παραπάνω παραδειγμάτων στο notepad++ και κάντε τους απαραίτητους ελέγχους. Δημιουργήστε ένα νέο αρχείο HTML και εισάγετε μια φόρμα εισαγωγής στοιχείων βιβλίων. Χρησιμοποιώντας το DOM τυπώστε το περιεχόμενο των στοιχείων της φόρμας και αλλάξτε το με κώδικα Javascript.

### Διαχείριση γεγονότων στην Javascript

Στις προηγούμενες ενότητες μάθαμε να δημιουργούμε προγράμματα σε Javascript και να τα εκτελούμε φορτώνοντας την ιστοσελίδα στον φυλλομετρητή. Η εκτέλεση αυτή γινόταν με την φόρτωση της ιστοσελίδας. Πολλές φορές όμως είναι απαραίτητο ενέργειες-κώδικας να εκτελούνται όταν υπάρχει διάδραση με τον χρήστη. Με άλλα λόγια είναι απαραίτητο να εκτελείται κώδικας σε Javascript όταν συμβαίνουν κάποια γεγονότα στην ιστοσελίδα. Τέτοια γεγονότα μπορεί να είναι το πάτημα ενός κουμπιού, το πάτημα σε ένα πλαίσιο κειμένου, η απομάκρυνση από ένα πλαίσιο κειμένου κ.α. Μερικά από τα πιο συνηθισμένα γεγονότα είναι:

- onclick, συμβαίνει όταν πατηθεί κουμπί
- onfocus, συμβαίνει όταν πατηθεί πλαίσιο κειμένου
- onblur, συμβαίνει όταν γίνει απομάκρυνση από ένα στοιχείο ελέγχου φόρμας
- onkeyup, συμβαίνει όταν αφήνουμε ένα πλήκτρο

Πριν προχωρήσουμε στην επεξήγηση το πώς η Javascript διαχειρίζεται αυτά τα γεγονότα, θα περιγράψουμε την λειτουργία της συνάρτησης στο Σχήμα 5.19. Η συνάρτηση αποτελεί μηχανισμό δημιουργίας υποπρογράμματος, έννοια που έχουμε συναντήσει και σε άλλα μαθήματα.

```
function welcome() {
    window.alert("Welcome to our library");
}

welcome();
```

*Σχήμα 5.19: Η συνάρτηση welcome()*

Για την δημιουργία μιας συνάρτησης στην Javascript γράφουμε την λέξη function. Ακολουθεί το όνομα που δίνουμε στην συνάρτηση και οι παρενθέσεις (). Εφόσον χρειάζονται παράμετροι, τις εισάγουμε μέσα στις παρενθέσεις(). Ακολουθεί το σώμα της συνάρτησης που βρίσκεται ανάμεσα στις αγκύλες {}. Για τυχόν επιστρεφόμενη τιμή από την συνάρτηση χρησιμοποιείται η λέξη return και η τιμή που θέλουμε η συνάρτηση να επιστρέψει. Για την κλήση της συνάρτησης γράφουμε το όνομα της συνάρτησης, τις παρενθέσεις () και τυχόν τιμές παραμέτρων μέσα σε αυτές.

Στο παράδειγμα του σχήματος 5.19 έχουμε την συνάρτηση `welcome()` που δεν έχει παραμέτρους και δεν επιστρέφει τιμή. Η μοναδική εντολή που υπάρχει στο σώμα της συνάρτησης είναι η:

**`window.alert("Welcome to our library");`**

Με αυτήν την εντολή δημιουργείτε ένα παράθυρο ειδοποίησης που εμφανίζει το μήνυμα "Welcome to our library". Η κλήση της συνάρτησης γίνεται με την εντολή **`welcome()`**;

```
<HTML>
  <HEAD>
    <TITLE>Βιβλιοθήκη</TITLE>
  </HEAD>
  <BODY>
    <H1 id="et1"> Our library site</h1>

    <FORM>
      ENTER YOUR NAME: <INPUT type="text" onfocus="welcome();" value="">
      <INPUT type="button" onclick="welcome();" value="OK">
    </FORM>

    <SCRIPT TYPE="text/javascript">
      function welcome() {
        window.alert("Welcome to our library");
      }
    </SCRIPT>

  </BODY>
</HTML>
```

**Σχήμα 5.20:** Κλήση συνάρτησης από γεγονός

Στο σχήμα 5.20 περιγράφεται μια ιστοσελίδα που περιέχει μια φόρμα. Η φόρμα αποτελείται από ένα πλαίσιο εισαγωγής κειμένου και ένα κουμπί. Στο πλαίσιο εισαγωγής κειμένου είναι εμφανής η ιδιότητα διαχείρισης γεγονότος `onfocus` η οποία παίρνει τιμή ίδια με το όνομα της συνάρτησης `welcome()`. Αυτό σημαίνει ότι όταν συμβεί το γεγονός πατήματος μέσα στο πλαίσιο κειμένου θα εκτελεστεί ο κώδικας της συνάρτησης `welcome()`. Ομοίως στο κουμπί έχουμε την ιδιότητα διαχείρισης γεγονότος `onclick` η οποία παίρνει τιμή ίδια με το όνομα της συνάρτησης `welcome()`. Αυτό σημαίνει ότι όταν συμβεί το γεγονός πατήματος του κουμπιού θα εκτελεστεί ο κώδικας της συνάρτησης `welcome()`. Στο παράδειγμα αυτό περιγράφηκε ο ποιο απλός τρόπος διαχείρισης γεγονότος ενσωματώνοντας τα γεγονότα ως ιδιότητες στις ετικέτες της HTML.

**Δραστηριότητα:** Εισάγετε τον κώδικα των παραπάνω παραδειγμάτων στο notepad++ και κάντε τους απαραίτητους ελέγχους.

## Παράδειγμα ελέγχου ορθής συμπλήρωσης φόρμα με Javascript

Σε αυτήν την ενότητα θα δούμε πως μπορούμε να χρησιμοποιήσουμε όλα αυτά που μάθαμε στην Javascript για να ελέγξουμε αν ο χρήστης έχει συμπληρώσει σωστά μια φόρμα εισαγωγής πληροφοριών βιβλίου στο σύστημα. Γίνεται η παραδοχή ότι ο έλεγχος αφορά την μη εισαγωγή κενής πληροφορίας σε κάθε πεδίο. Στο πρώτο βήμα σχεδιάζεται η φόρμα εισαγωγής στοιχείων στην HTML όπως φαίνεται στο σχήμα 5.21. Κάθε στοιχείο της φόρμας (πλαίσια κειμένου) έχει εισαχθεί σε ένα πίνακα. Επίσης έχει προστεθεί και μια τρίτη στήλη στον πίνακα (κενή προς το παρόν) για την προβολή μηνυμάτων-οδηγιών στον χρήστη ανά πεδίο εισαγωγής. Αξιοσημείωτο είναι ότι σε όλα τα πλαίσια κειμένου αλλά και στα κενά πεδία της τρίτης στήλης του πίνακα έχει προστεθεί η ιδιότητα `id` (για να μπορούν να είναι εύκολα και διαχειρίσιμα από το DOM). Όσον αφορά τα πλαίσια κειμένου έχει προστεθεί και η ιδιότητα-γεγονός `onkeyup` (όταν αφήνουμε ένα πλήκτρο) στην οποία εκχωρείται η συνάρτηση `checking()`;

```
<HTML>
<HEAD>
  <TITLE>Βιβλιοθήκη</TITLE>
</HEAD>
<BODY>
  <H1 id="et1"> Our library site</h1>

  <FORM>
    <TABLE border="1">
      <tr><td>Book Title</td><td><INPUT type="text" id="book_t" onkeyup="checking();" value=""></td><td id="comments_t"></td></tr>
      <tr><td>Book ISBN</td><td><INPUT type="text" id="book_i" onkeyup="checking();" value=""></td><td id="comments_i"></td></tr>
      <tr><td>Book Author</td><td><INPUT type="text" id="book_a" onkeyup="checking();" value=""></td><td id="comments_a"></td></tr>
    </TABLE>
    <INPUT type="button" id="button1" value="OK">
  </FORM>
</BODY>
</HTML>
```

**Σχήμα 5.21:** Φόρμα εισαγωγής βιβλίων

Το επόμενο βήμα είναι να γραφεί κώδικας σε Javascript (για το γεγονός όταν αφήνουμε πλήκτρο) που θα ελέγχει κάθε πλαίσιο κειμένου. Αν το πλαίσιο κειμένου είναι κενό θα εμφανίζεται στο διπλανό κελί του πίνακα μήνυμα, που θα προτρέπει το χρήστη να εισάγει κατάλληλη τιμή. Διαφορετικά θα εμφανίζεται το μήνυμα “OK”. Ο κώδικας αυτός εμφανίζεται στο σχήμα 5.22.

```

<SCRIPT TYPE="text/javascript">
function checking() {
var elements = new Array();
var displays = new Array();
var messages = new Array();

elements[0] = document.getElementById("book_t").value;
elements[1] = document.getElementById("book_i").value;
elements[2] = document.getElementById("book_a").value;

displays[0] = "comments_t";
displays[1] = "comments_i";
displays[2] = "comments_a";

messages[0] = "Please insert book title";
messages[1] = "Please insert book ISBN";
messages[2] = "Please insert book Author";

    for (var i = 0; i < elements.length; i++){
        if (elements[i]==""){
            document.getElementById(displays[i]).innerHTML = messages[i];
        }else{
            document.getElementById(displays[i]).innerHTML = "OK";
        }
    }

}
</SCRIPT>

```

*Σχήμα 5.22: Κώδικας ελέγχου ορθότητας δεδομένων φόρμας*

Με το DOM η ιδιότητα value του κάθε πλαισίου κειμένου εισάγεται σε ένα πίνακα (elements). Στον πίνακα displays εισάγονται τα id των κελιών του πίνακα όπου θα εμφανίζονται τα μηνύματα ενώ στον πίνακα messages αποθηκεύονται τα μηνύματα. Κατόπιν για κάθε πλαίσιο κειμένου (for) γίνεται έλεγχος αν ο χρήστης δεν έχει εισάγει δεδομένα (elements[i] == ""). Αν δεν έχει εισάγει, μέσω DOM

document.getElementById(display[i]).innerHTML=messages[i];

εκχωρείται στο κατάλληλο κελί το κατάλληλο μήνυμα. Αν έχει εισάγει, με παρόμοιο τρόπο εκχωρείται η τιμή "OK".

**Δραστηριότητα:** Εισάγετε τον κώδικα των παραπάνω παραδειγμάτων στο notepad++ και κάντε τους απαραίτητους ελέγχους. Αναζητήστε στο διαδίκτυο συναρτήσεις διαχείρισης αλφαριθμητικών έτσι ώστε να δημιουργήσετε μια νέα φόρμα εισαγωγής στοιχείων μαθητή πάνω στην οποία θα γίνεται έλεγχος ορθότητας δεδομένων δίνοντας έμφαση στην ορθότητα του εισαγόμενου e-mail.

## 5.4 Προγραμματισμός εξυπηρετητή σε PHP

Η **PHP** είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που μπορεί να κάνει όλα τα είδη των πραγμάτων, όπως να συλλέγει και να διαχειρίζεται τα δεδομένα μιας φόρμας που αποστέλλεται από ένα πρόγραμμα περιήγησης, να δημιουργεί προσαρμοσμένο διαδικτυακό (web) περιεχόμενο για να εξυπηρετήσει το πρόγραμμα περιήγησης, να επικοινωνεί με μια βάση δεδομένων, να περιορίζει τους χρήστες ώστε αυτοί να έχουν πρόσβαση σε συγκεκριμένες σελίδες στην ιστοσελίδα σας καθώς και να στέλνει και να λαμβάνει cookies (δηλαδή μικρά πακέτα δεδομένων που χρησιμοποιεί το πρόγραμμα περιήγησής σας για να θυμάται ορισμένα πράγματα, όπως για παράδειγμα αν είστε συνδεδεμένοι στο Πανελλήνιο Σχολικό Δίκτυο).

Η PHP είναι γλώσσα συγγραφής σεναρίων (scripting language) για δημιουργία δυναμικού διαδικτυακού περιεχομένου. Τα σενάρια (scripts ) PHP εκτελούνται στον διακομιστή (server) και όχι στον πελάτη (client). Μέσα σε μία HTML σελίδα μπορείτε να ενσωματώσετε PHP κώδικα που θα εκτελείται κάθε φορά που θα επισκέπτεστε την συγκεκριμένη σελίδα. Γενικά, σε κώδικα HTML, CSS και JavaScript παρεμβάλλεται κώδικας PHP. Τα αρχεία PHP έχουν κατάληξη .php και αποθηκεύονται και εκτελούνται στον διακομιστή web. Ο PHP κώδικας μεταφράζεται στον διαδικτυακό διακομιστή (web server) και παράγεται περιεχόμενο σε μορφή κώδικα HTML ή άλλο περιεχόμενο που ο χρήστης μπορεί να δει.

Ρίξτε μια ματιά στον παρακάτω κώδικα. Σας θυμίζει κάτι; Πολλοί θα αναγνωρίσετε στον κώδικα αυτόν την γλώσσα HTML. Και πού ακριβώς βρίσκεται ο κώδικας PHP; Ο κώδικας PHP γράφεται μεταξύ των διαχωριστικών-οριοθετών (tags) `<?php` και `?>`. Μπορείτε να μαντέψετε τι ακριβώς κάνει ο κώδικας αυτός; Μήπως εμφανίζει ένα μήνυμα στην οθόνη; Και πιο είναι αυτό; Μαντέψατε σωστά. Το μήνυμα είναι το εξής «Θέλω να μάθω να προγραμματίζω σε PHP». Ποια συνάρτηση χρησιμοποιεί για να εμφανίσει το μήνυμα αυτό; Το βρήκατε. Είναι η συνάρτηση echo.

```
<html>
  <head>
</head>
  <body>
    <p>
      <?php
        echo "Θέλω να μάθω να προγραμματίζω σε PHP";
      ?>
    </p>
  </body>
</html>
```

Θα μπορούσατε να ρωτήσετε το εξής: Το ίδιο πράγμα θα μπορούσα να το κάνω και σε JavaScript. Αυτό ισχύει. Η διαφορά έγκειται στο γεγονός ότι με την JavaScript μπορώ να κάνω λιγότερα πράγματα από ότι με την PHP. Ας εξηγήσουμε το γιατί.



Η JavaScript τρέχει γενικά στο πρόγραμμα περιήγησης ή αλλιώς στον πελάτη (client). Αυτό σημαίνει ότι γνωρίζει πραγματικά μόνο δύο πράγματα. Αυτά που συμβαίνουν στο πρόγραμμα περιήγησης σας καθώς και αυτά που του στέλνει η ιστοσελίδα με την οποία συνδέεστε. Η PHP από την άλλη πλευρά τρέχει στον ίδιο υπολογιστή (ή αλλιώς στον διακομιστή (server)) με τον ιστότοπο που επισκέπτεστε. Αυτό σημαίνει ότι έχει πρόσβαση σε όλες τις πληροφορίες και τα αρχεία που υπάρχουν σε αυτόν τον υπολογιστή. Η παραπάνω πρόσβαση δίνει στην PHP την δυνατότητα να δημιουργήσει προσαρμοσμένες σελίδες HTML για να τις στείλει στη συνέχεια στο πρόγραμμα περιήγησης σας, να χειριστεί τα cookies καθώς και να εκτελέσει διάφορες εργασίες χρησιμοποιώντας δεδομένα που βρίσκονται στον διακομιστή.

**Δραστηριότητα:** Τροποποιήστε τον παραπάνω κώδικα για να εμφανίσετε το όνομά σας.

**Δραστηριότητα:** Μαντέψτε τι θα εμφανιστεί με το παρακάτω τμήμα κώδικα:

```
<?php
    echo 5 * 7;
?>
```

Για να αρχίσετε να αναπτύσσετε διαδικτυακές εφαρμογές PHP που συνομιλούν με βάσεις δεδομένων χρειάζεστε έναν διακομιστή web server (για παράδειγμα XAMPP) ακόμα και στον δικό σας υπολογιστή, όπου μπορείτε να εγκαταστήσετε την PHP, την MySQL καθώς και έναν απλό επεξεργαστή κειμένου (για παράδειγμα notepad++). Ενδεικτικό παράδειγμα τοπικής εγκατάστασης xampp υπάρχει στο παράρτημα των σημειώσεων. Εναλλακτική λύση παρέχουν και οι on line πλατφόρμες ελέγχου κώδικα PHP σε περίπτωση που δεν θέλουμε να εγκαταστήσουμε λογισμικό.

#### 5.4.1 Μεταβλητές

Μέχρι τώρα έχουμε εμφανίσει λέξεις και αποτελέσματα πράξεων. Για να δημιουργήσουμε πιο σύνθετα προγράμματα, χρειαζόμαστε έναν τρόπο για να «σώσουμε» αυτές τις τιμές. Μπορούμε να το κάνουμε αυτό χρησιμοποιώντας μεταβλητές. Μια μεταβλητή μπορεί να αποθηκεύσει μία λέξη ή έναν αριθμό. Για παράδειγμα, με την παρακάτω δήλωση

**\$myAge = 32**

αποθηκεύουμε στην μεταβλητή με το όνομα \$myAge τον αριθμό 32. Όλα τα ονόματα των μεταβλητών αρχίζουν υποχρεωτικά με το σύμβολο \$. Μετά το σύμβολο \$ μπορεί να ακολουθούν λατινικοί χαρακτήρες (A - z), ψηφία (0 - 9) ή ο χαρακτήρας «\_» με οποία σειρά θέλουμε. Τα ονόματα των μεταβλητών είναι case-sensitive, δηλ. το \$myAge και το \$myage είναι δύο διαφορετικές μεταβλητές.

**Δραστηριότητα:** Στον παρακάτω κώδικα δημιουργήστε δύο μεταβλητές με το όνομα \$myName και \$myAge και εκχωρήστε σε αυτές το όνομά σας και την ηλικία σας

αντίστοιχα. Εμφανίστε την τιμή των παραπάνω μεταβλητών χρησιμοποιώντας την εντολή echo. Τοποθετήστε τις εντολές εκχώρησης τιμής μεταξύ των tags <?php και ?>

```
<html>
  <head>
  </head>
  <body>
    <p>
      <?php

      ?>
    </p>
  </body>
</html>
```

Οι τύποι δεδομένων που υποστηρίζονται από την PHP είναι οι εξής: String, Integer, Float (λέγεται και double), Boolean, Array, Object, NULL και Resource.

Το String είναι μία ακολουθία από χαρακτήρες, όπως "Hello world!" ανάμεσα όμως σε απλά ή διπλά εισαγωγικά.

**Δραστηριότητα:** Εκτελέσετε τον παρακάτω κώδικα για να μάθετε τι αυτοκίνητο οδηγεί ο James Bond. Μπορείτε να αλλάξετε το αυτοκίνητο ή/και τον οδηγό; Το σύμβολο // το χρησιμοποιούμε για να γράφουμε σχόλια. Τι καινούριο μάθατε από την εκτέλεση αυτού του κώδικα;

```
<?php
  $identity = 'James Bond';
  $car = 'BMW';
  // Η μεταβλητή $sentence μας λέει ποιος οδηγεί και τι αυτοκίνητο
  $sentence = "$identity drives a $car";
  echo $sentence;
?>
```

**Δραστηριότητα:** Πόσοι τρόποι υπάρχουν να πούμε το ίδιο πράγμα; Εκτελέστε τον παρακάτω κώδικα για να μάθετε τι εννοούμε. Τι καινούριο μάθατε από την εκτέλεση αυτού του κώδικα;

```
<?php
  // αυτό...
```

```
$a = 5;

$a = $a + 10;

// ... είναι ίδιο με αυτό

$a = 5;

$a += 10;

?>
```

**Δραστηριότητα:** Ας επανέλθουμε πάλι στα String. Πόσοι τρόποι υπάρχουν να πούμε το ίδιο πράγμα; Εκτελέσετε τον παρακάτω κώδικα για να μάθετε τι εννοούμε. Τι καινούριο μάθατε από την εκτέλεση αυτού του κώδικα; Ποιος είναι ο ρόλος της τελείας «.»;

```
<?php

// αυτό...

echo "Hello, world!";

// ... είναι ίδιο με αυτό

echo "Hello," . " " . "world" . "!";

?>
```

#### 5.4.2 Δομή Επιλογής

Ας θεωρήσουμε ότι θέλουμε να γράψουμε ένα πρόγραμμα το οποίο να ρωτάει αν η ηλικία σου είναι μεγαλύτερη ή ίση του 18. Αν η απάντηση είναι ναι, θέλουμε να εμφανίζεται ένα μήνυμα που να λέει «Μπορείς να ψηφίσεις!». Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε την δομή επιλογής **if**, όπως φαίνεται στο παρακάτω παράδειγμα:

```
<?php

    $age = 19;
    if( $age >= 18 ) {
        echo " Μπορείς να ψηφίσεις!";
    }

?>
```

Στο παραπάνω παράδειγμα, χρησιμοποιήσαμε την εντολή **if** για να κάνουμε κάτι, να εμφανίσουμε δηλαδή ένα μήνυμα, αν η απάντηση στην ερώτηση είναι ναι δηλαδή **Αληθής** όπως λέμε στην γλώσσα προγραμματισμού PHP. Τι γίνεται στην περίπτωση που θέλουμε να εμφανίσουμε ένα μήνυμα που να λέει «Είσαι ακόμα μικρός!», όταν η απάντηση στην ερώτηση είναι όχι; Χρησιμοποιούμε τότε την σύνθετη δομή επιλογής **if / else** όπως φαίνεται στο παρακάτω παράδειγμα:

```
<?php
    $age = 17;
    if( $age >= 18 ) {
        echo " Μπορείς να ψηφίσεις!";
    }
    else {
        echo " Είσαι ακόμα μικρός!";
    }
?>
```

### 5.4.3 Δομή Επανάληψης

Όταν δημιουργούμε προγράμματα, συχνά πρέπει να επαναλάβουμε τα ίδια πράγματα πολλές φορές, όπως στο παρακάτω παράδειγμα που πρέπει να εμφανίσουμε αριθμούς από το 1 μέχρι και το 100. Πόσες φορές πρέπει να καλέσουμε την εντολή echo; Ας σημειώσουμε ότι το σύμβολο «//» στην αρχή της φράσης «και τα λοιπά και τα λοιπά» χρησιμοποιείται για να εισάγουμε σχόλια στο πρόγραμμά μας. Τα σχόλια είναι μόνο για εμάς και μας βοηθούν να θυμόμαστε τι κάνει το πρόγραμμά μας.

```
<?php
    echo 1;
    echo 2;
    echo 3;
    // και τα λοιπά και τα λοιπά
?>
```

Υπάρχει τρόπος να γράψουμε λιγότερες εντολές; Ας μελετήσουμε το παρακάτω παράδειγμα στο οποίο χρησιμοποιείται η δομή επανάληψης «for»:

```
<?php
    // Εμφανίστε τους αριθμούς από το 1 μέχρι και το 100
    for ($i = 1; $i < 101; $i = $i + 1) {
        echo $i;
    }
?>
```

Η δομή επανάληψης «for» ξεκινάει με την δεσμευμένη λέξη «for». Είναι σαν να λέμε στην PHP ετοιμάσου να εκτελέσεις αυτό που θα σου πούμε πολλές φορές. Στη συνέχεια προσθέτουμε ένα ζευγάρι από παρενθέσεις «()». Μέσα στις παρενθέσεις «()» λέμε στην PHP τρία πράγματα. Τα τρία αυτά πράγματα χωρίζονται μεταξύ τους με το σύμβολο «;» και λένε αντίστοιχα τα εξής: Πώς αρχίζει η επανάληψη, πότε τελειώνει η επανάληψη και τι κάνουμε κάθε φορά πριν πάμε στην επόμενη επανάληψη. Στη συνέχεια των παρενθέσεων ακολουθούν τα σύμβολα «{}». Μέσα στα σύμβολα «{}» λέμε στην PHP ποιές εντολές να εκτελεί σε κάθε επανάληψη.

Έτσι, το παραπάνω τμήμα κώδικα λέει στην PHP τα εξής: Ξεκίνα τον κύκλο των επαναλήψεων θέτοντας στην μεταβλητή \$i την τιμή 1, σταμάτα τον κύκλο των επαναλήψεων πριν φτάσει η μεταβλητή \$i την τιμή 101, αύξανε κάθε φορά την τιμή της μεταβλητής \$i κατά 1 πριν πάς στην επόμενη επανάληψη και σε κάθε επανάληψη εμφάνιζε την τρέχουσα τιμή της μεταβλητής \$i.

**Δραστηριότητα:** Στο παρακάτω πρόγραμμα προσπαθώ να εμφανίσω τους πέντε πρώτους άρτιους αριθμούς. Δηλαδή τους 2, 4, 6, 8, και 10. Κάτι όμως δεν κάνω καλά. Μπορείτε να βρείτε τι φταίει;

```
<html>
<head>
  <title>Δομή επανάληψης For </title>
</head>
<body>
  <p>
    <?php
      // Εμφάνισε τους πέντε πρώτους άρτιους αριθμούς
      for ($i = 2; $i < 10; $i = $i + 1){
        echo $i;
      }
    ?>
  </p>
</body>
</html>
```

Όταν θέλουμε να επαναλάβουμε ένα σύνολο εντολών πολλές φορές χωρίς όμως να γνωρίζουμε εκ των προτέρων τον αριθμό των επαναλήψεων τότε χρησιμοποιούμε την εντολή **while**. Θα ρωτήσετε πότε σταματάμε τον κύκλο των επαναλήψεων στην περίπτωση αυτή; Ο κύκλος των επαναλήψεων τερματίζει όταν μία συγκεκριμένη συνθήκη παύει να ισχύει, δηλαδή όταν γίνει ψευδής (false). Μελετήστε το παρακάτω παράδειγμα:

```
<?php
$loopCond = true;
while ($loopCond){
  echo "<p> Είμαστε μέσα στην επανάληψη </p>";
  $loopCond = false;
```

```

}
echo "<p> Βγήκαμε από την επανάληψη </p>";
?>

```

Η δομή επανάληψης «while» συντάσσεται με τον ακόλουθο τρόπο:

```

while(condition) {
    // εντολές που εκτελούνται κάθε φορά όσο η συνθήκη condition είναι αληθής
}

```

Με τις επαναλήψεις, πρέπει να είμαστε προσεκτικοί όσον αφορά τον τερματισμό τους. Θεωρείται ότι ο παρακάτω κύκλος επαναλήψεων θα τερματίσει κάποια στιγμή;

```

while(2 > 1){
    // εντολές
}

```

**Δραστηριότητα:** Ποια εντολή πρέπει να προσθέσουμε για να τερματίσει ο κύκλος των επαναλήψεων; Θυμηθείτε ότι ο Ο κύκλος των επαναλήψεων τερματίζει όταν η συνθήκη που συνοδεύει το while παύει να ισχύει, δηλαδή όταν γίνει ψευδής (false).

```

<?php
    $loopCond = true;
    while ($loopCond == true ){
        echo "<p> Η επανάληψη τρέχει. </p>";
        // Πρόσθεσε μία εντολή που θα σταματήσει την επανάληψη
    }
    echo ".. και τώρα τελείωσε";
?>

```

Έχετε παρατηρήσει ότι η επανάληψη while ελέγχει πρώτα την συνθήκη που την συνοδεύει και στην περίπτωση που είναι αληθής εκτελείται ο κώδικας που περιέχεται μεταξύ των αγκύλων { }. Μία εναλλακτική περίπτωση της while αποτελεί η **do/while** που κάνει ακριβώς το αντίθετο. Δηλαδή, πρώτα εκτελείται η επανάληψη και στη συνέχεια ελέγχεται η συνθήκη.

**Δραστηριότητα:** Μελετήστε το παρακάτω παράδειγμα και εξηγήστε γιατί η εντολή «echo \$i» θα εκτελεστεί μόνο μία φορά.

```

<?php

```

```

$i = 0;

do {
    echo $i;
} while ($i > 0);

?>

```

#### 5.4.4. Πίνακες

Ένας πίνακας είναι μια λίστα από αντικείμενα, δηλαδή σαν μία λίστα αγορών. Σας επιτρέπει να αποθηκεύσετε περισσότερα από ένα αντικείμενα σε μία μόνο μεταβλητή.

Σκεφτείτε το εξής. Όταν γράφετε μία λίστα με τα ψώνια σας, θα μπορούσατε να χρησιμοποιήσετε ένα ξεχωριστό φύλλο χαρτί για κάθε αντικείμενο που θέλετε να αγοράσετε. Το κάθε φύλλο χαρτί είναι και μια ξεχωριστή μεταβλητή. Μπορείτε να φανταστείτε τον εαυτό σας με τόσα χαρτιά στο χέρι όσα και τα πράγματα που θέλετε να αγοράσετε; Γιατί να μην χρησιμοποιήσετε ένα φύλλο χαρτί για όλα αυτά που θέλετε να αγοράσετε; Αυτό το φύλλο χαρτί το οποίο περιέχει όλα αυτά που θέλετε να αγοράσετε ονομάζεται **πίνακας (array)**. Στο παρακάτω παράδειγμα, ο πίνακας με όνομα \$array περιέχει τα στοιχεία «ντομάτες», «αυγά» και «ζάχαρη».

```

<html>

<head>

    <title>Πίνακες!</title>

</head>

<body>

    <?php

        $array = array("ντομάτες", "αυγά", "ζάχαρη");

    ?>

</body>

</html>

```

**Δραστηριότητα:** Ποιο στοιχείο του πίνακα θα εμφανιστεί στο παρακάτω παράδειγμα;

```

<?php

    $myArray = array("do", "re", "mi");

    echo $myArray[0]

?>

```

**Δραστηριότητα:** Εξηγείστε τι συμβαίνει στο παρακάτω τμήμα κώδικα.

```
<?php
    $languages = array("HTML/CSS", "JavaScript", "I don't know", "Python",
    "Ruby");
    $languages[2] = "PHP";
    echo $languages[2] ;
?>
```

#### 5.4.5. Συναρτήσεις

Οι συναρτήσεις είναι επαναχρησιμοποιήσιμα κομμάτια κώδικα που μπορείτε να χρησιμοποιήσετε μέσα στο πρόγραμμά σας. Η PHP έχει πολλές ενσωματωμένες συναρτήσεις. Μία από αυτές είναι η `strlen()` που επιστρέφει τον αριθμό των χαρακτήρων που υπάρχουν σε μία λέξη (ή πιο σωστά σε ένα αλφαριθμητικό). Στο παρακάτω παράδειγμα εμφανίζεται ο αριθμός 5, όσοι είναι και οι χαρακτήρες της λέξης «Μαρία».

```
<?php
    $length = strlen("Μαρία");
    print $length;
?>
```

Εκτός από τις ενσωματωμένες συναρτήσεις PHP που μας παρέχονται μπορούμε να δημιουργήσουμε και τις δικές μας, όπως φαίνεται στο παρακάτω παράδειγμα, όπου εμφανίζουμε το όνομά μας.

```
<?php
    function displayname() {
        echo "Το όνομά μου είναι Πέτρος";
    }
    displayname();
?>
```

Οι συναρτήσεις δεν θα ήταν τόσο χρήσιμες αν δεν μπορούσαν να δεχτούν κάποια δεδομένα, ας τα ονομάσουμε δεδομένα εισόδου (input), τα οποία θα μπορούσαν στη συνέχεια να τα επεξεργαστούν. Ας ονομάσουμε τα δεδομένα εισόδου παράμετροι (parameters) ή ορίσματα (arguments). Οι παράμετροι είναι μεταβλητές, τις τιμές των οποίων χρησιμοποιεί η συνάρτηση για να κάνει υπολογισμούς.



Στο παρακάτω τμήμα κώδικα η συνάρτηση δέχεται μία παράμετρο, την οποία την πολλαπλασιάζει με τον εαυτό της και στη συνέχεια εμφανίζει το αποτέλεσμα της αριθμητικής πράξης.

```
<?php
    function squareValue($number) {
        echo $number * $number;
    }
    $n = 6;
    // Εμφανίζεται ο αριθμός 36
    squareValue($n);
?>
```

**Δραστηριότητα:** Μπορείτε να εξηγήσετε τι κάνει το παρακάτω πρόγραμμα;

```
<?php
    function aboutMe($name, $age) {
        echo "Hello! My name is " . $name . " and I am " . $age . " years old.";
    }
    aboutMe( "Maria", 17);
?>
```

**Δραστηριότητα:** Γράψτε τον κώδικα μίας συνάρτησης την οποία θα ονομάσετε greetings. Η συνάρτηση αυτή θα δέχεται ως παράμετρο το \$name και θα εμφανίζει με την βοήθεια του echo ένα string το οποίο θα είναι ίσο με "Greetings, " . \$name . "!". Στη συνέχεια καλέστε την συνάρτηση greetings() με το όνομά σας για να δοκιμάσετε τι έχετε κάνει.

**Δραστηριότητα:** Εξηγείστε τι συμβαίνει στο παρακάτω τμήμα κώδικα. Ποιος είναι ο ρόλος του return σε σχέση με το echo;

```
<?php
    function sum($a, $b) {
        $c = $a + $b;
        return $c;
    }
```

```
echo sum(2,13) ;
```

```
?>
```

#### 5.4.6. Πρόσβαση και διαχείριση Βάσεων Δεδομένων MySQL από το διαδίκτυο χρησιμοποιώντας PHP

Ο συνδυασμός της PHP με την βάση δεδομένων MySQL και με την υποστήριξη ενός διαδικτυακού εξυπηρετητή (web server) μας επιτρέπει να δημιουργήσουμε διαδικτυακές εφαρμογές προσαρμοσμένες στις ανάγκες των χρηστών. Ας δούμε την λειτουργία της αρχιτεκτονικής των διαδικτυακών βάσεων δεδομένων, μέσω ενός παραδείγματος αυτό της αναζήτησης βιβλίων.

- Το πρόγραμμα περιήγησης ενός χρήστη κάνει μία http:// αίτηση για μια συγκεκριμένη ιστοσελίδα. Ο χρήστης αναζητεί τα βιβλία ενός συγκεκριμένου συγγραφέα χρησιμοποιώντας μία HTML φόρμα. Συμπληρώνει μία φόρμα με τα στοιχεία που αναζητεί και κάνει κλικ στο κουμπί «Υποβολή». Στην περίπτωση αυτή καλείται για παράδειγμα ένα σενάριο που ονομάζεται results.php.
- Ο διαδικτυακός εξυπηρετητής (web server) λαμβάνει ένα αίτημα για το results.php. Βρίσκει το αρχείο αυτό και το δίνει στην μηχανή (engine) PHP για επεξεργασία.
- Η μηχανή (engine) PHP αρχίζει να αναλύει το σενάριο. Μέσα στο σενάριο περιέχονται εντολές σύνδεσης με την βάση δεδομένων και εκτέλεσης του ερωτήματος (αναζήτηση των βιβλίων). Η PHP ανοίγει μία σύνδεση με τον εξυπηρετητή MySQL και στέλνει το ερώτημα.
- Ο εξυπηρετητής MySQL λαμβάνει το ερώτημα, το επεξεργάζεται και στέλνει πίσω στην μηχανή PHP τα αποτελέσματα, δηλαδή την λίστα με τα βιβλία.
- Η μηχανή PHP μορφοποιεί κατάλληλα τα αποτελέσματα σε μορφή κώδικα HTML και τα επιστρέφει στον διαδικτυακό εξυπηρετητή.
- Ο διαδικτυακός εξυπηρετητής τα στέλνει στη συνέχεια στο πρόγραμμα περιήγησης, όπου ο χρήστης μπορεί να δει την λίστα με τα βιβλία που ζήτησε.

Παρακάτω εμφανίζονται τα βασικά τμήματα του περιεχόμενου του σεναρίου results.php που ανακτά τα αποτελέσματα αναζήτησης από την βάση δεδομένων MySql και τα μορφοποιεί για παρουσίαση.



#### 5.4.7. Προβολή δεδομένων βάσης δεδομένων MySQL σε ιστοσελίδα HTML

Βασική λειτουργία της PHP αποτελεί η διασύνδεση της με βάση δεδομένων (MySQL) με σκοπό την προβολή τους σε ιστοσελίδα HTML. Στο παράδειγμα μας, στην απομακρυσμένη βάση δεδομένων library υπάρχει ο πίνακας books με τα πεδία btitle, isbn και bauthor. Σκοπός του παραδείγματος (σχήμα 5.23) είναι η προβολή των στοιχείων όλων των βιβλίων που είναι αποθηκευμένα στην βάση σε ιστοσελίδα HTML.

```

1  <?php
2
3  //Προβολή δεδομένων από απομακρυσμένη βάση MySQL σε ιστοσελίδα HTML
4  //Σύνδεση με την βάση δεδομένων
5  $hostname="localhost"; //εναλλακτικά την διεύθυνση του server που έχει την Mysql
6  $username="root";
7  $password="yourpassword";
8  $dbname="library";
9
10 $conn=mysql_connect($hostname, $username, $password);
11 mysql_select_db($dbname,$conn);
12
13 echo '<h3>Τα βιβλία μας,</h3>';
14 $result = mysql_query('SELECT * FROM books',$conn) or die('cannot show tables');
15 if(mysql_num_rows($result)) {
16     echo '<table cellpadding="0" cellspacing="0" class="db-table">';
17     echo '<tr><th>Τίτλος</th><th>ISBN</th><th>Συγγραφέας</th></tr>';
18     while($row = mysql_fetch_row($result)) {
19         echo '<tr>';
20         for (i=0; i<3; i++) {
21             echo '<td>',$row[i],</td>';
22         }
23         echo '</tr>';
24     }
25     echo '</table><br />';
26
27 }
28 ?>

```

*Σχήμα 5.23: Προβολή δεδομένων σε HTML από MySQL*

Αφού γίνει σύνδεση με τον απομακρυσμένο υπολογιστή (mysql\_connect εντολές 5,6,7,10) επιλέγεται η βάση δεδομένων library (mysql\_select\_db εντολές 8, 11). Κατόπιν χρησιμοποιείται η εντολή echo προκειμένου να εισαχθεί στην ιστοσελίδα (μέσω PHP) η ετικέτα HTML<H3>. Στη συνέχεια υποβάλλεται το ερώτημα επιλογής όλων των δεδομένων του πίνακα books (mysql\_query εντολή 14). Εφόσον το ερώτημα επιστρέψει δεδομένα (εντολή 15 – έλεγχος αν \$result είναι διαφορετικό του null) δημιουργείται η κεφαλίδα του πίνακα προβολής (εντολή echo 16,17). Η επανάληψη while (εντολή 18) στον πίνακα \$row φορτώνει τα δεδομένα κάθε γραμμής (σε κάθε επανάληψη) του πίνακα books. Το εμφωλευμένο for (εντολές 20, 21, 22) τυπώνει σε κελί πίνακα HTML το κάθε πεδίο της γραμμής (\$row) του πίνακα books.

#### 5.4.8. Εισαγωγή Δεδομένων φόρμας σε Βάση Δεδομένων MySQL από το διαδίκτυο χρησιμοποιώντας PHP

Η αντίστροφη λειτουργία με αυτήν που περιγράφηκε στην προηγούμενη ενότητα είναι αυτή της εισαγωγής δεδομένων από φόρμα HTML και αποθήκευση τους σε απομακρυσμένη βάση δεδομένων MySQL. Στην απομακρυσμένη βάση δεδομένων library υπάρχει ο πίνακας books με τα πεδία btitle, isbn και bauthor. Στο πρώτο βήμα δημιουργείται φόρμα HTML όπως φαίνεται στο σχήμα 5.24.

```

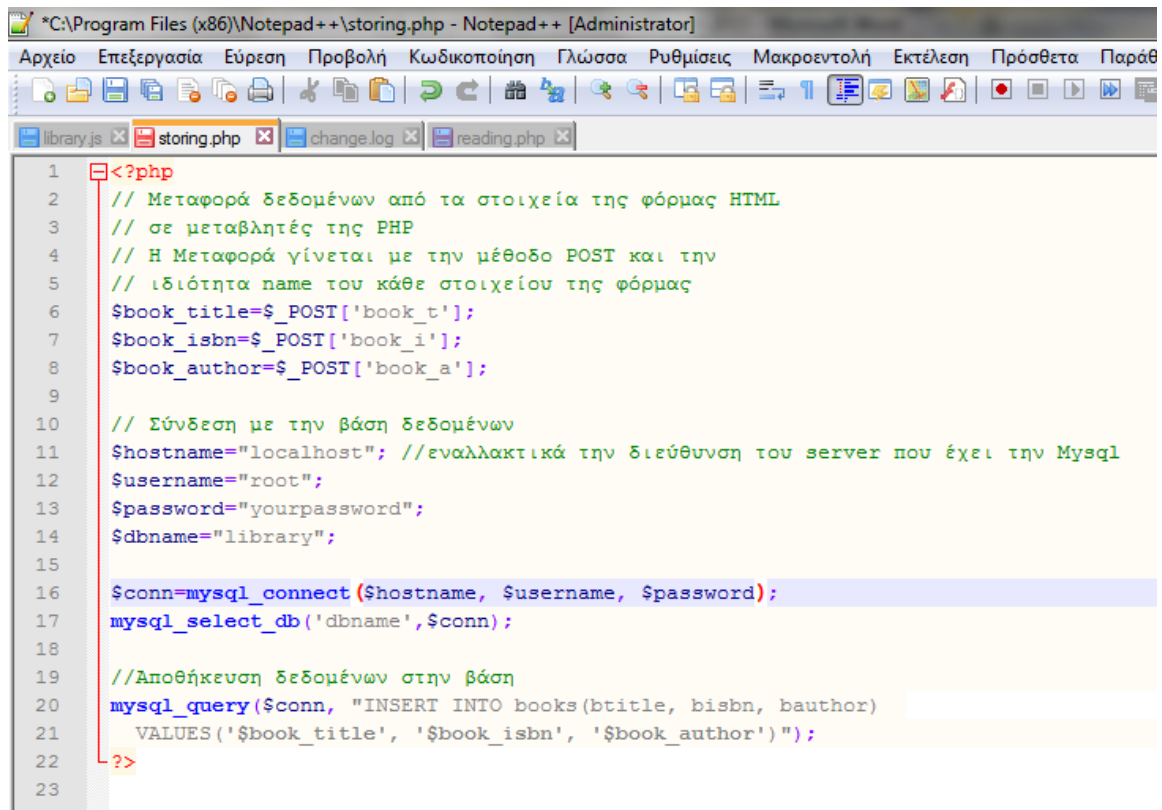
<HTML>
  <HEAD>
    <TITLE>Βιβλιοθήκη</TITLE>
  </HEAD>
  <BODY>
    <H1> Εισαγωγή βιβλίου</H1>

    <FORM action="storing.php" method="post">
      Book Title <INPUT type="text" name="book_t" value=""><br />
      Book ISBN <INPUT type="text" name="book_i" value=""><br />
      Book Author <INPUT type="text" name="book_a" value=""><br />
      <INPUT type="submit" name="submit" value="submit">
    </FORM>
  </BODY>
</HTML>

```

**Σχήμα 5.24:** Φόρμα εισαγωγής βιβλίων

Στην ιδιότητα action της φόρμας εκχωρείται το όνομα του αρχείου PHP (storing.php) του οποίου ο κώδικας θα εκτελεστεί όταν πατηθεί το πλήκτρο “submit”. Με την ιδιότητα method δηλώνεται ο τρόπος μεταφοράς δεδομένων (POST) από την HTML στην PHP. Ο κώδικας PHP που αποθηκεύει τα δεδομένα σε μία απομακρυσμένη βάση δεδομένων περιγράφεται στο σχήμα 5.25.



```

1  <?php
2  // Μεταφορά δεδομένων από τα στοιχεία της φόρμας HTML
3  // σε μεταβλητές της PHP
4  // Η Μεταφορά γίνεται με την μέθοδο POST και την
5  // ιδιότητα name του κάθε στοιχείου της φόρμας
6  $book_title=$_POST['book_t'];
7  $book_isbn=$_POST['book_i'];
8  $book_author=$_POST['book_a'];
9
10 // Σύνδεση με την βάση δεδομένων
11 $hostname="localhost"; //εναλλακτικά την διεύθυνση του server που έχει την Mysql
12 $username="root";
13 $password="yourpassword";
14 $dbname="library";
15
16 $conn=mysql_connect($hostname, $username, $password);
17 mysql_select_db($dbname, $conn);
18
19 //Αποθήκευση δεδομένων στην βάση
20 mysql_query($conn, "INSERT INTO books(btitle, bisbn, bauthor)
21     VALUES('$book_title', '$book_isbn', '$book_author')");
22
23

```

*Σχήμα 5.25: Κώδικας PHP αποθήκευσης δεδομένων*

Αρχικά αποθηκεύονται οι τιμές από την φόρμα HTML μέσω του POST σε μεταβλητές της PHP. Κατόπιν γίνεται η σύνδεση στην απομακρυσμένη βάση δεδομένων και υποβάλλεται το ερώτημα αποθήκευσης.

#### 5.4.9. Σύνοδοι/Sessions στην PHP

Στα προηγούμενα παραδείγματα μάθαμε το πως αποθηκεύονται και το πώς ανακτώνται δεδομένα σε/από μια βάση δεδομένων. Λειτουργίες πολύ χρήσιμες για την υλοποίηση ενός πληροφοριακού συστήματος. Παρόλα αυτά, δεν επαρκούν (με τον τρόπο με τον οποίο περιγράφηκαν στις προηγούμενες ενότητες) καθώς αυτές οι λειτουργίες είναι διαθέσιμες για όλους τους χρήστες του διαδικτύου. Όταν όμως κάνουμε είσοδο (login) στον e-mail λογαριασμό μας και μετά, το περιεχόμενο είναι διαθέσιμο μόνο για εμάς μολονότι δουλεύοντας με το e-mail μας πλοηγούμαστε σε διαφορετικές σελίδες του ίδιου ιστότοπου. Με άλλα λόγια υπάρχει η ανάγκη μεταφοράς δεδομένων (ταυτότητα χρήστη) από σελίδα σε σελίδα.

Οι σύνοδοι είναι μεταβλητές με τις οποίες μπορούν να μεταφερθούν δεδομένα από σελίδα σε σελίδα. Σε αντίθεση με τα cookies (αποθήκευση δεδομένων χρήστη στον φυλλομετρητή) οι σύνοδοι (sessions) αποθηκεύονται στον εξυπηρετητή (server). Για την δημιουργία sessions χρησιμοποιείται η εντολή:

**session\_start();**

Ενώ για την αποθήκευση δεδομένων χρησιμοποιείται ο πίνακας \$\_SESSION. Ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει όσες θέσεις θέλει σε αυτόν τον πίνακα. Ως δείκτες μπορεί να χρησιμοποιήσει όποια ονομασία επιθυμεί και φυσικά να εισάγει οποιαδήποτε τιμή όπως φαίνεται παρακάτω:

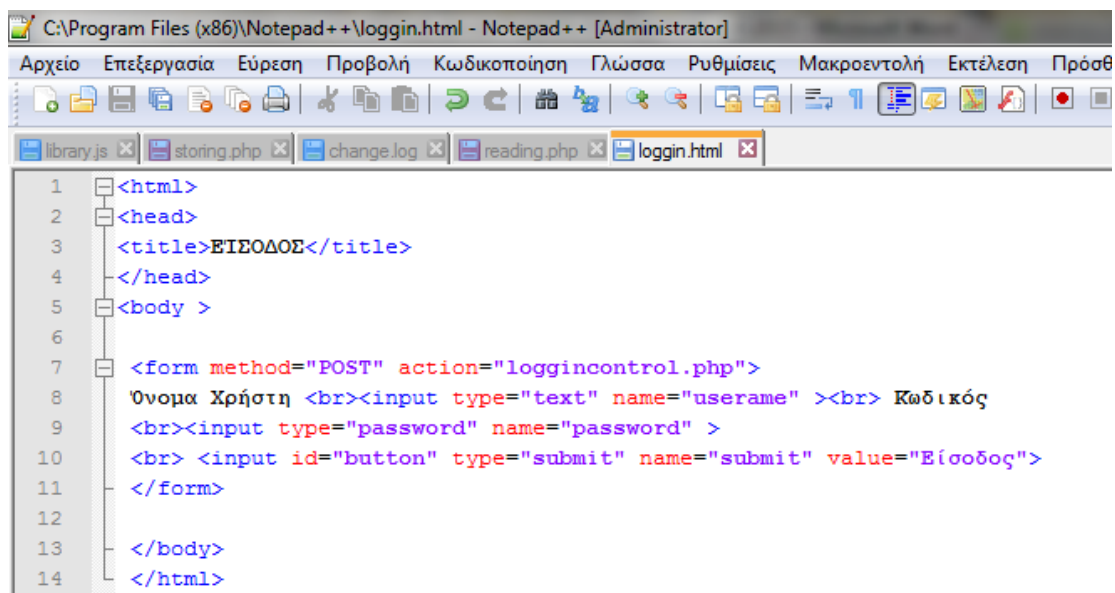
```
$_SESSION["loggedIn"]="true";
```

```
$_SESSION["username"]="someuser";
```

Οποιαδήποτε επόμενη σελίδα εκκινήσει το session (session\_start() ) μπορεί να διαβάσει οποιαδήποτε θέση του πίνακα \$\_SESSION με τον κατάλληλο δείκτη (π.χ. \$\_SESSION["username"]) ή να αλλάξει την τιμή οποιασδήποτε θέσης με εκχώρηση. Για να κλείσει το session (π.χ. logout) πρέπει μια σελίδα να καλέσει την εντολή:

```
session_destroy();
```

Στο παράδειγμα που ακολουθεί δημιουργείται ιστοσελίδα στην οποία ο χρήστης εισάγει σε φόρμα το όνομα χρήστη και τον κωδικό του, γίνεται έλεγχος ορθής εισόδου με την βάση δεδομένων και σε περίπτωση επιτυχίας δημιουργείται session με το οποίο στην επόμενη σελίδα εμφανίζει το όνομα χρήστη. Αρχικά δημιουργούμε το αρχείο login.html (σχήμα 5.26) όπου εμφανίζεται μια φόρμα με τα στοιχεία εισόδου και όταν πατηθεί το κουμπί submit εκτελείται ο κώδικας στο αρχείο logincontrol.php (σχήμα 5.27).



*Σχήμα 5.26: Φόρμα στοιχείων εισόδου*

```

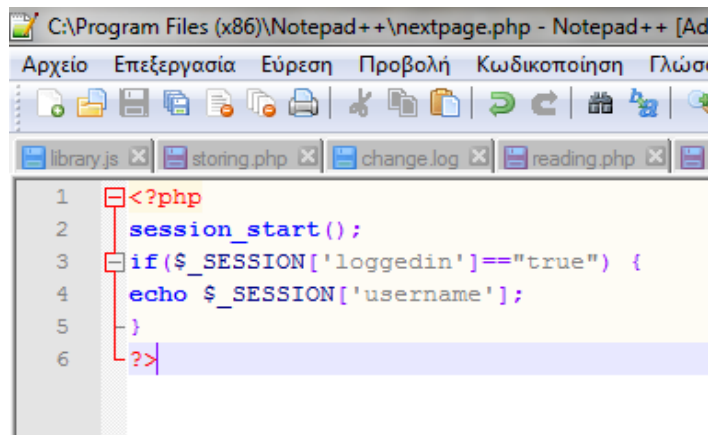
1 <?php
2 $username=$_POST['username'];
3 $password=$_POST['password'];
4
5 //Σύνδεση με την βάση δεδομένων
6 $hostname="localhost"; //εναλλακτικά την διεύθυνση του server που έχει την Mysql
7 $username="root";
8 $password="yourpassword";
9 $dbname="library";
10
11 $conn=mysql_connect($hostname, $username, $password);
12 mysql_select_db($dbname,$conn);
13
14 $query = mysql_query("SELECT * FROM users where username = '$username' AND password = '$password'") or die(mysql_error());
15
16 if($query) {
17     session_start();
18     $_SESSION['loggedin']="true";
19     $_SESSION['username']=$username;
20     echo "Καλώς ορίσατε..";
21     header('Location: nextpage.php');
22 }
23 else { echo "Λάθος στοιχεία εισόδου.."; }
24
25 -?>

```

**Σχήμα 5.27:** Έλεγχος στοιχείων εισόδου

Μέσω post μεταφέρονται τα δεδομένα εισόδου του χρήστη στην PHP. Αφού γίνει σύνδεση στη βάση, γίνεται ερώτηση στον πίνακα users για εγγραφή με όνομα χρήστη και κωδικό ίδιο με αυτό που εισήγαγε ο χρήστης. Σε περίπτωση επιτυχίας δημιουργείται το session με τις γνωστές μεταβλητές και μεταφέρεται η ροή στην σελίδα nextpage.php (σχήμα 5.28) όπου γίνεται έλεγχος στην \$\_SESSION['loggedin'] και αν είναι "true" τυπώνεται το όνομα χρήστη μέσω της αντίστοιχης session μεταβλητής.



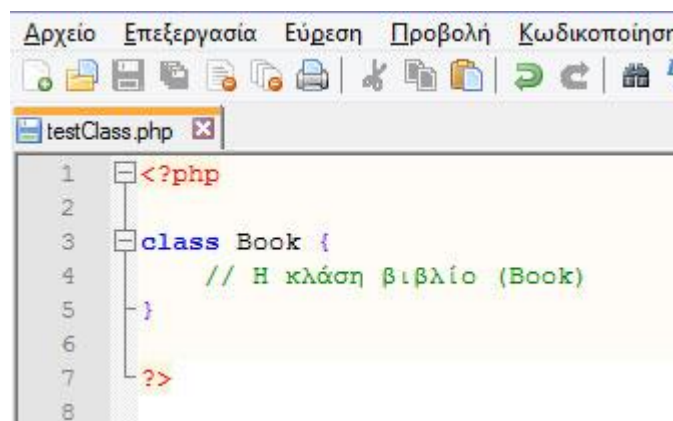


Σχήμα 5.28:nextpage.php

#### 5.4.10. Δημιουργία κλάσεων στην php

Στο 3<sup>ο</sup> κεφάλαιο παρουσιάστηκε η αντικειμενοστραφής προσέγγιση ανάλυσης απαιτήσεων και στο 4<sup>ο</sup> κεφάλαιο ο σχεδιασμός προσανατολισμένος στα αντικείμενα. Στη φάση της υλοποίησης δημιουργούνται οι κλάσεις στη γλώσσα προγραμματισμού που έχει επιλεγεί για να αναπτυχθεί το σύστημα.

Η διαδικασία για να δημιουργηθεί μια κλάση στην γλώσσα προγραμματισμού php είναι αρκετά απλή. Αρχικά, συνήθως σε ένα νέο αρχείο (π.χ. το αρχείο testClass.php), δηλώνουμε την κλάση γράφοντας την δεσμευμένη λέξη **class**, το όνομα της κλάσης και τις αγκύλες {}:



Η κλάση αποτελεί το πρότυπο βάσει του οποίου δημιουργούνται τα αντικείμενα κατά τον χρόνο εκτέλεσης ενός προγράμματος. Στην php μπορούμε δημιουργήσουμε ένα αντικείμενο κατά τον χρόνο εκτέλεσης ενός προγράμματος και να το αποθηκεύσουμε σε μια μεταβλητή. Για να δημιουργήσουμε ένα αντικείμενο π.χ. τύπου «Book» χρησιμοποιούμε την δεσμευμένη λέξη **new** ως εξής:

```
$book = new Book;
```

Όπως έχει αναφερθεί στο κεφάλαιο 4.3 οι κλάσεις αποτελούνται από τις ιδιότητες και τις μεθόδους. Οι ιδιότητες δηλώνονται μέσα στην κλάση ως μεταβλητές:

```
class Book {
    var $title = "Η ιδιότητα τίτλος!!!";
}
```

Οι μέθοδοι δηλώνονται μέσα στην κλάση ως συναρτήσεις:

```
class Book {
    var $title = "Η ιδιότητα τίτλος!!!";

    function getTitle() {
        // Η μέθοδος πάρει ή δες Τίτλο!!!
    }
}
```

Οι ιδιότητες και οι μέθοδοι που περιέχονται σε κάθε αντικείμενο προσεγγίζονται από το αντικείμενο με τη χρήση του βέλους (->):

```
echo $bookObj->getTitle();
```

Για να αναφερθούμε στη μέθοδο ή στις ιδιότητες μιας κλάσης μέσα από την ίδια την κλάση χρησιμοποιούμε τη λέξη **\$this**:

```
class Book {
    var $title = "Η ιδιότητα τίτλος!!!";

    function getTitle() {
        return $this->title;
    }

    function setTitle($t) {
        // Η μέθοδος θέσε Τίτλο!!!
        $this->title = $t;
    }
}
```

Εκτός απ' τις μεθόδους και τις μεταβλητές που έχουν αποτυπωθεί στη φάση του σχεδιασμού, όλες οι κλάσεις περιλαμβάνουν εξ' ορισμού κάποιες μεθόδους όπως είναι η μέθοδος constructor (κατασκευαστής). Η μέθοδος constructor καλείται κάθε φορά που δημιουργείται ένα νέο αντικείμενο και έχει το **όνομα της κλάσης** ή το όνομα **\_\_construct**. Η μέθοδος constructor μπορεί να παίρνει σαν είσοδο ένα ή περισσότερα ορίσματα:

```

class Book {
    var $title;

    function Book($t) {
        $this->title = $t;
    }

    function getTitle() {
        return $this->title;
    }

    function setTitle($t) {
        $this->title = $t;
    }
}

```

Για να δημιουργηθούν αντικείμενα της παραπάνω κλάσης δίνονται σαν είσοδο τα κατάλληλα δεδομένα που ορίζονται στη μέθοδο constructor:

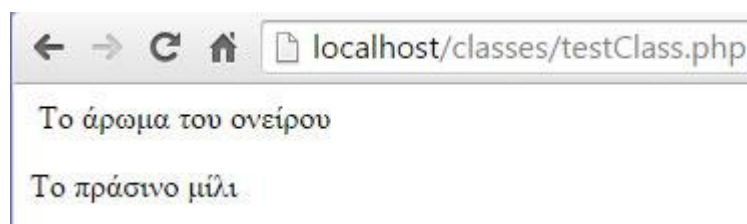
```

$book1Obj = new Book("Το άρωμα του ονείρου");
$book2Obj = new Book("Το πράσινο μίλι");

echo $book1Obj->getTitle() . "<p>";
echo $book2Obj->getTitle();

```

Στο πρόγραμμα περιήγησή σας πληκτρολογήστε την διεύθυνση του αρχείου testClass.php:



**Δραστηριότητα:** Να δημιουργήσετε την κλάση Book με τις κατάλληλες ιδιότητες και μεθόδους.

#### 5.4.11. Αντικείμενα και βάσεις δεδομένων

Τα αντικείμενα μπορούν να συνδέονται με τη βάση δεδομένων, να ανακτούν δεδομένα, να αποθηκεύουν ή να διαγράφουν δεδομένα. Ένα παράδειγμα MySQL πίνακα για να αποθηκεύονται τα βιβλία στη βάση δεδομένων φαίνεται παρακάτω:

```

CREATE TABLE IF NOT EXISTS `book` (
  `aa` varchar(10) NOT NULL DEFAULT '0',
  `title` varchar(150) NOT NULL DEFAULT "",
  `writer` varchar(100) NOT NULL DEFAULT "",

```

```

`publisher` varchar(100) NOT NULL DEFAULT "",
`edition` varchar(100) NOT NULL DEFAULT "",
`no_of_pages` smallint(4) NOT NULL DEFAULT '0',
`category` varchar(10) NOT NULL DEFAULT '0',
`cover` varchar(100) NOT NULL DEFAULT 'default.png',
`review` mediumtext NOT NULL,
PRIMARY KEY `aa` (`aa`),
KEY `idx_cat` (`category`),
KEY `idx_aa` (`aa`),
KEY `idx_writer` (`writer`)
);

```

όπου ‘aa’ είναι ο αύξων αριθμός που δίνει ο υπεύθυνος της βιβλιοθήκης στο βιβλίο, το ‘category’ είναι η κατηγορία που ανήκει το βιβλίο π.χ. λογοτεχνικό, επιστημονικό κτλ, ‘cover’ είναι το όνομα της εικόνας – εξώφυλλο του βιβλίου και ‘review’ είναι μια μικρή περίληψη για το περιεχόμενο του βιβλίου.

Με τη δημιουργία ενός αντικειμένου Book θα πρέπει να γίνει εισαγωγή των κατάλληλων δεδομένων στον πίνακα. Έτσι θα πρέπει να δημιουργηθεί ένα πεδίο στον παραπάνω πίνακα με τη χρήση της εντολής **INSERT**:

```

INSERT INTO book (`aa`, `title`, `writer`, `publisher`, `edition`, `no_of_pages`,
`category`, `cover`, `review`) VALUES ('1', 'Ματωμένα χρώματα', 'Διδώ Σωτηρίου',
'Κέδρος', '2008', 344, 'Λογοτεχνία', 'matomena_xomata.jpg', '«Πόλεμοι και ξανά
πόλεμοι! Τι στο καλό θα βγάλει η εποχή μας...');

```

Η συγκεκριμένη εντολή **INSERT** θα πρέπει να καλείται από τη μέθοδο constructor ή από οποιαδήποτε άλλη μέθοδο δημιουργηθεί γι’ αυτό τον σκοπό όπως π.χ. μια μέθοδος storeBook. Η μέθοδος αυτή θα πρέπει να παίρνει σαν είσοδο όλα τα δεδομένα που χρειάζονται για να δημιουργηθεί η καταχώρηση στη βάση:

```

function Book($aa, $t, $w, $p, $e, $nop, $cat, $co, $r) {

    $book_iquery_str = "INSERT INTO book (`aa`, `title`, `writer`, `publisher`,
`edition`, `no_of_pages`, `category`, `cover`, `review`) VALUES ('$aa', '$t', '$w', '$p',
'$e', '$nop', '$cat', '$co', '$r');"

    $book_iquery = mysql_query($book_iquery_str) or die (mysql_error());
}

```

Η εντολή **mysql\_query** στέλνει ερωτήματα SQL στη βάση δεδομένων και επιστρέφει μια λίστα με καταχωρήσεις που επιστρέφονται από το ερώτημα. Υπενθυμίζεται ότι

πριν γίνει ένα ερώτημα προς τη βάση θα πρέπει να έχει γίνει σύνδεση με τον εξυπηρετητή MySQL και επιλογή της κατάλληλης βάσης δεδομένων.

Αντίστοιχα η εντολή `setTitle` θα πρέπει να καλεί ένα ερώτημα **UPDATE** προς τη βάση για να τροποποιηθεί ο τίτλος του βιβλίου που έχει αποθηκευτεί στη βάση:

```
function setTitle($t) {
    $uquery_str = "UPDATE `book` SET title='$t' WHERE aa='$this->aa'";
    $uquery = mysql_query($uquery_str) or die (mysql_error());
    $this->title = $t;
}
```

Επίσης είναι απαραίτητη μια μέθοδος που ανακτά τα δεδομένα από τη βάση δεδομένων στις ιδιότητες ενός αντικειμένου όπως π.χ. το `retrieveBook`. Η μέθοδος μπορεί να δημιουργηθεί με τη χρήση της εντολής **SELECT**:

```
function retrieveBook($aa) {
    $this->aa = $aa;
    $book_squery_str = "SELECT * FROM book WHERE aa='$aa'";
    $book_squery = mysql_query($book_squery_str) or die (mysql_error());
    $row = mysql_fetch_array ($book_squery);
    $this->title = $row['title'];
    $this->writer = $row['writer'];
    $this->publisher = $row['publisher'];
    $this->edition = $row['edition'];
    $this->no_of_pages = $row['no_of_pages'];
    $this->category = $row['category'];
    $this->cover = $row['cover'];
    $this->review = $row['review'];
}
```

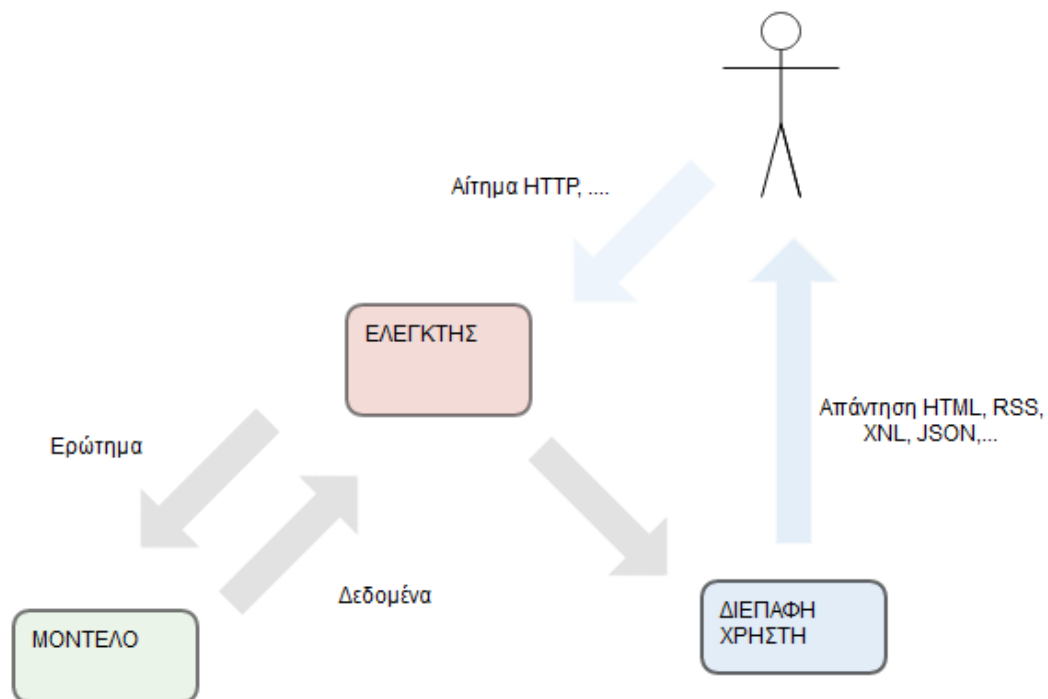
Η εντολή **mysql\_fetch\_array** επιστρέφει έναν πίνακα με τα αποτελέσματα του ερωτήματος SQL προς τη βάση.

**Δραστηριότητα:** Να εμπλουτίσετε την κλάση `Book` με τις κατάλληλες μεθόδους που θα διατυπώνουν ερωτήματα προς τον πίνακα `Book` της βάσης δεδομένων.

#### 5.4.12. MVC Πλαίσια για την δημιουργία διαδικτυακών PHP εφαρμογών (PHP Web Applications)

Η PHP είναι μια σταθερή πλατφόρμα, αλλά είναι ακόμα πιο ισχυρή όταν χρησιμοποιούμε ένα από τα πλαίσια MVC (Model View Controller), όπως το CakePHP, το Zend, το Symfony και το CodeIgniter που μας επιτρέπουν να συντομεύσουμε τον χρόνο ανάπτυξης και να οργανώσουμε καλύτερα τον κώδικά μας. Το πλαίσιο MVC αποτελεί το σκελετό γύρω από τον οποίο κτίζεται η εφαρμογή μας. Το πρότυπο MVC όπως φαίνεται στο παρακάτω σχήμα 5.9 χωρίζει μία εφαρμογή σε τρία συνεκτικά μέρη: στο μοντέλο (model), στη διεπαφή χρήστη (view) και στον ελεγκτή (controller).

Το μοντέλο (Model) είναι υπεύθυνο για την αποθήκευση και την ανάκτηση των δεδομένων. Αποτελεί εκείνο το μέρος της εφαρμογής που περιλαμβάνει την διαχείριση των δεδομένων. Η διεπαφή χρήστη (View) είναι υπεύθυνη για το πώς τα δεδομένα αυτά θα παρουσιαστούν στον χρήστη. Αποτελεί εκείνο το μέρος της εφαρμογής που περιλαμβάνει την παραγωγή κώδικα HTML και την αλληλεπίδραση με το χρήστη. Ο ελεγκτής χειρίζεται τα αιτήματα του πελάτη σε συνεργασία με το μοντέλο (Model) και την διεπαφή χρήστη (View). Δέχεται το αίτημα του χρήστη, το επεξεργάζεται, το αναλύει και στη συνέχεια στέλνει το αντίστοιχο ερώτημα στο μοντέλο (Model). Η απάντηση που θα πάρει από το μοντέλο (Model) θα την προωθήσει στη διεπαφή χρήστη (View) για την κατάλληλη μορφοποίηση.



*Σχήμα 5.25: Ένα βασικό MVC αίτημα*

### **Ερωτήσεις – Δραστηριότητες – Θέματα προς συζήτηση**

1. Η τεχνολογία ενισχύει την ικανότητα των εκπαιδευτικών ιδρυμάτων και ινστιτούτων να προσφέρουν εκπαιδευτικό περιεχόμενο σε μορφή λιγότερο ή

περισσότερο δομημένη σε όλους ανεξάρτητα από γεωγραφικά σύνορα. Χαρακτηριστικά παραδείγματα των παραπάνω αποτελούν το Coursera, το edX, το Khan Academy και το YouTube EDU. Επισκεφτείτε τους παραπάνω ιστότοπους και παρουσιάστε στην τάξη τους στόχους τους, το είδος του εκπαιδευτικού περιεχομένου καθώς και το πώς θα μπορούσαν να χρησιμοποιηθούν για την υποστήριξη των παραδοσιακών τρόπων παροχής της εκπαίδευσης.

2. Στον τομέα της υγείας, από την πλευρά του ασθενούς, πώς θα νιώθατε σε περίπτωση που χρειαζόνταν να γίνει διάγνωση από έναν γιατρό ο οποίος θα βρίσκονταν εκατοντάδες ή χιλιάδες χιλιόμετρα μακριά από εσάς; Τι θα περιμένατε ή τι θα σας προβλημάτιζε περισσότερο αν βρισκόσασταν στην θέση αυτή;

### ***Βιβλιογραφία***

- O'Brien J.A & Marakas G.(2008). *Introduction to Information Systems*, 14th/edition, McGraw-Hill, New York.
- Quickly E. & Gargenta M. (2009). *PHP and MySQL by Example, 1st Edition*, Prentice Hall, Boston, MA.
- McPeak J. & Wilton P. (2015). *Beginning JavaScript*, 5<sup>th</sup> edition, WROX.
- Robson E. & Freeman E. (2005). *Head first HTML with CSS and XHTML*, O'Reilly Media.
- Satzinger J.W., Jackson R.B. & Burd S.D (2009). *Systems Analysis and Design in a Changing World*, Boston, MA: Thomson Course Technology.
- Shklar L. & Rosen R (2003). *Web Application Architecture, Principles, protocols and practices*. John Wiley & Sons, England.

### ***Διαδικτυακές Πηγές***

- HTML The language for building web pages  
<http://www.w3schools.com/default.asp>
- Intro to HTML/CSS: Making webpages  
<https://www.khanacademy.org/computing/computer-programming/html-css>
- Learn how to create websites by structuring and styling your pages with HTML and CSS <http://www.codecademy.com/en/tracks/web>
- Learn to program in PHP <http://www.codecademy.com/learn>
- PHP 5 tutorial <http://www.w3schools.com/php/>
- PHP Documentation [http:// php.net](http://php.net)

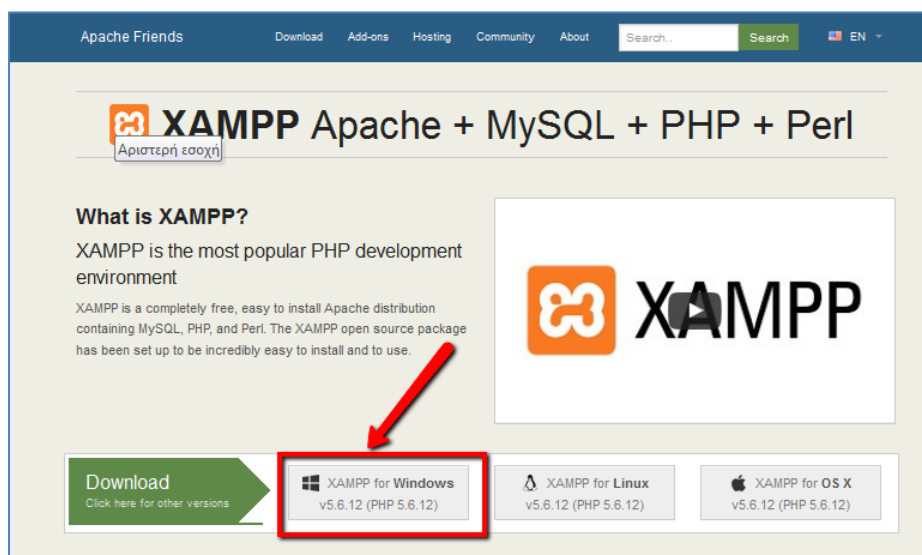
## ΠΑΡΑΡΤΗΜΑ

Στο παράρτημα αυτό παρέχονται γενικές οδηγίες για το πώς μπορούμε να εγκαταστήσουμε τοπικά php (γλώσσα προγραμματισμού για δημιουργία δυναμικών ιστοσελίδων), mysql (βάση δεδομένων), apache (web server) και phpmyadmin (εργαλείο διαχείρισης της MySQL) με τη βοήθεια του xampp. Παρουσιάζονται επιπλέον, απλά προγράμματα σε PHP διαχείρισης βάσεων δεδομένων, όπως σύνδεση με την βάση δεδομένων, δημιουργία πίνακα, εισαγωγή δεδομένων και εμφάνιση δεδομένων πίνακα. Τέλος, παρατίθενται ενδεικτικά παραδείγματα διαχείρισης βάσεων δεδομένων μέσω phpmyadmin. Ας σημειώσουμε, ότι μπορούμε να εγκαταστήσουμε τα συστατικά του xampp μεμονωμένα, ξεχωριστά το ένα από το άλλο.

### Εγκατάσταση xampp

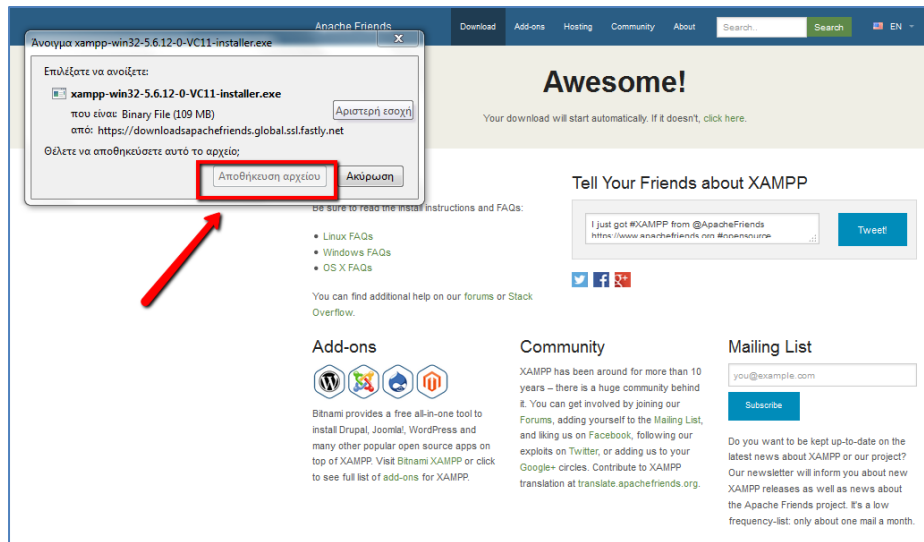
Το XAMPP είναι ένα δωρεάν πρόγραμμα για διάφορα λειτουργικά συστήματα που δημιουργεί ένα περιβάλλον ενός τοπικού διακομιστή (Server) στον υπολογιστή μας, ο οποίος μπορεί να μεταφράσει δεδομένα της γλώσσας προγραμματισμού PHP και της γλώσσας βάσεων δεδομένων. Το XAMPP είναι ιδανικό για τις δοκιμές μας, χωρίς αυτό να σημαίνει ότι δεν θα μπορούσαμε να αναπτύξουμε παρόμοιες εφαρμογές και για ένα απομακρυσμένο διακομιστή καθώς οι περισσότεροι web διακομιστές υποστηρίζουν τα ίδια συστατικά με αυτά του xampp.

Μεταβαίνουμε αρχικά στην ιστοσελίδα <https://www.apachefriends.org>

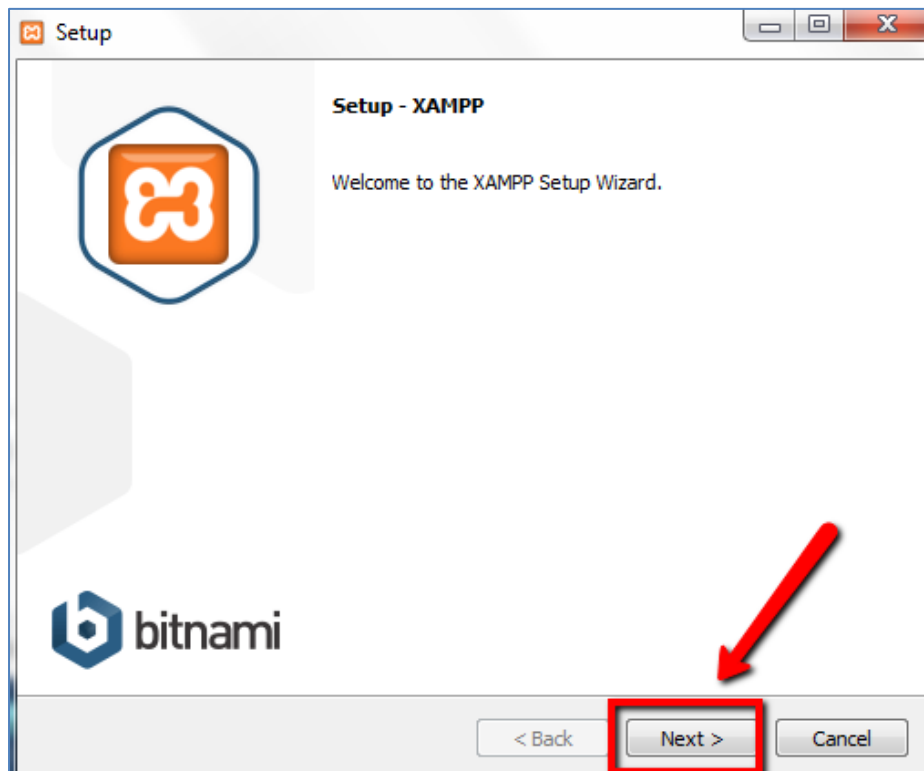


Κατεβάζουμε εκείνη την έκδοση του xampp σύμφωνα με το λειτουργικό μας σύστημα.

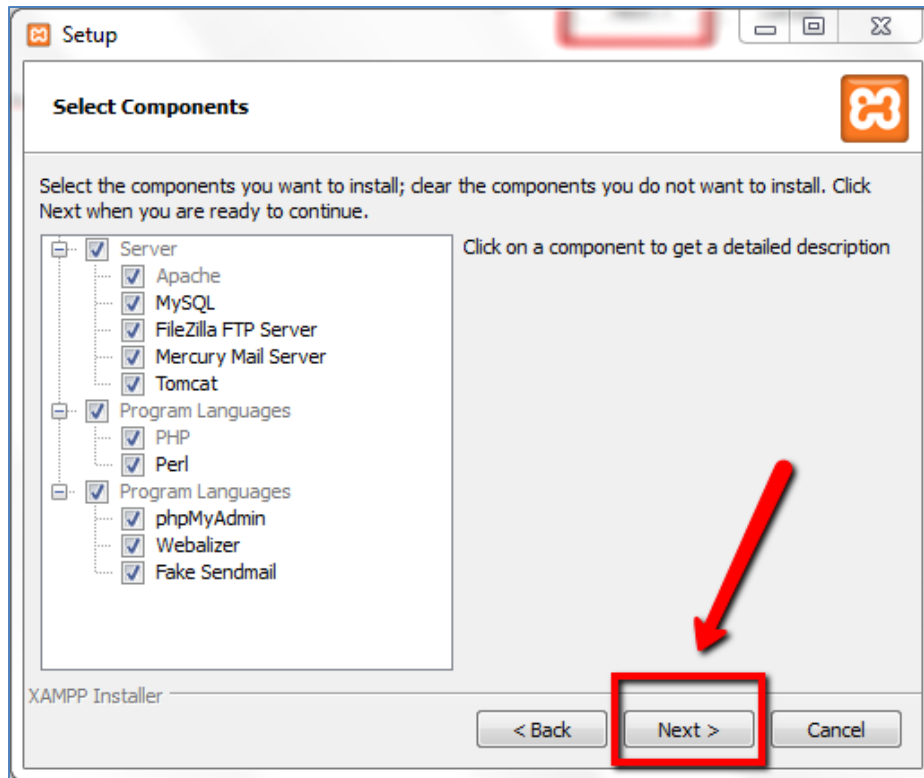




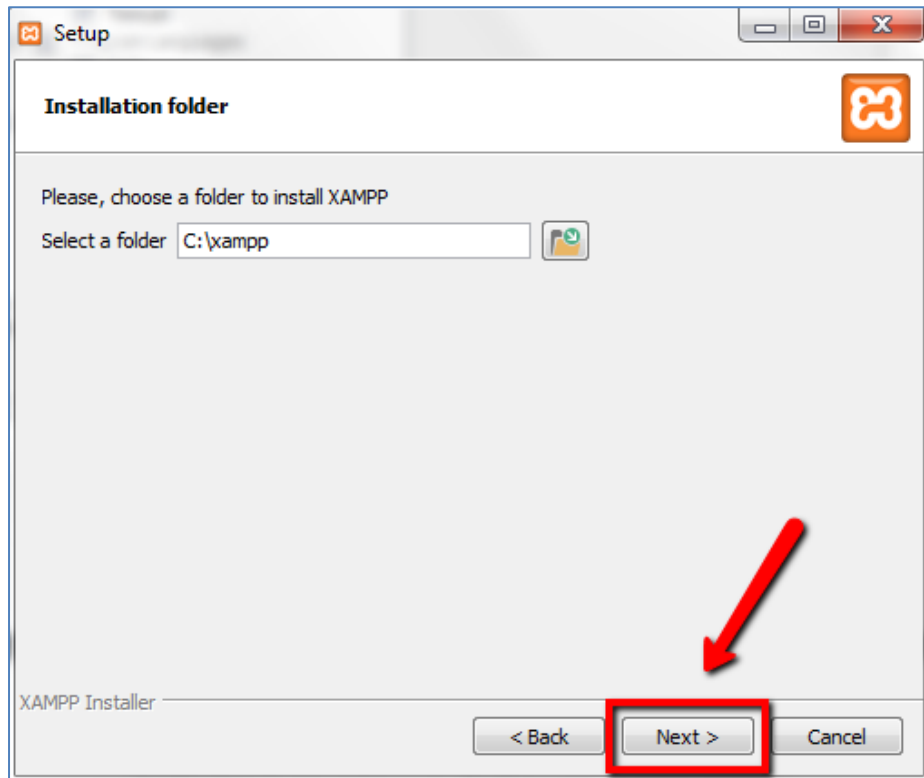
Αποθηκεύουμε και εγκαθιστούμε το αντίστοιχο αρχείο .exe. Κατά την εγκατάσταση είναι δυνατό να εμφανιστούν διάφορα μηνύματα, όπως να κλείσουμε το antivirus και άλλα. Πατάμε «Yes» στα αντίστοιχα πλαίσια διαλόγου για να συνεχιστεί η εγκατάσταση.



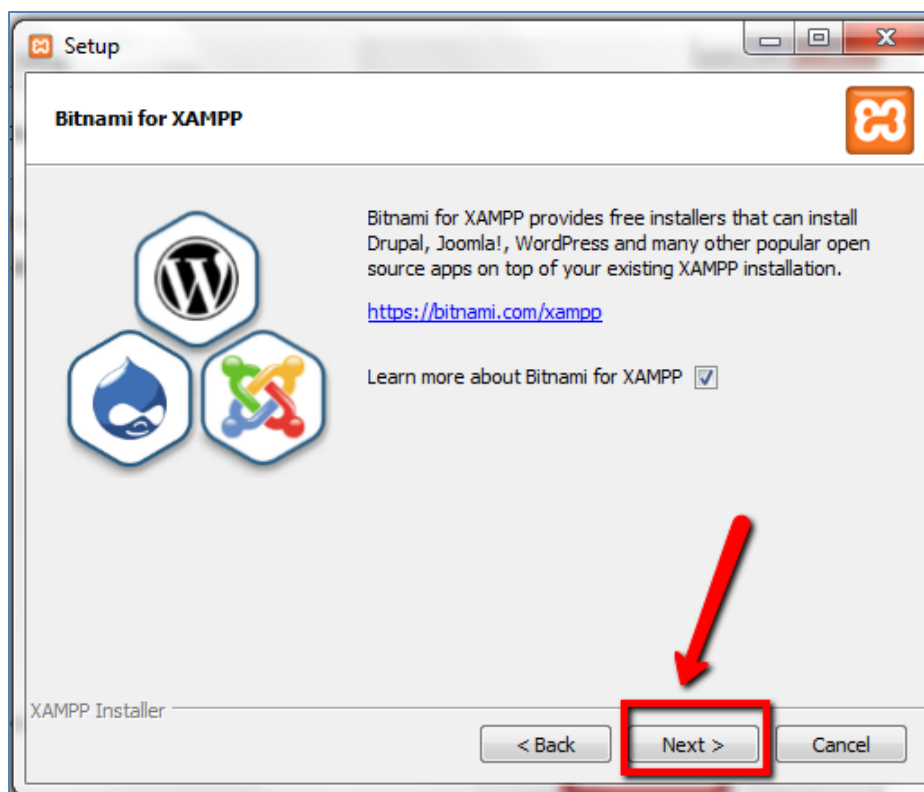
Παραμετροποιούμε την εγκατάσταση σύμφωνα με τις ανάγκες μας.



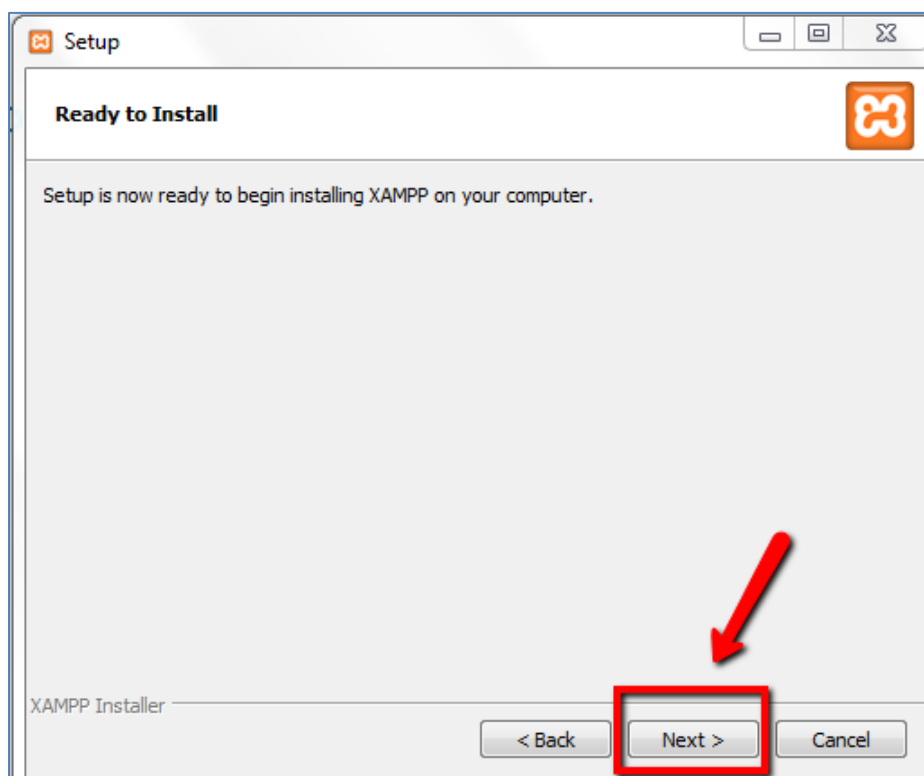
Το πρόγραμμά μας θα εγκατασταθεί στον φάκελο c:\xampp. Η τοποθεσία εγκατάστασης μπορεί να αλλάξει αν το επιθυμούμε.

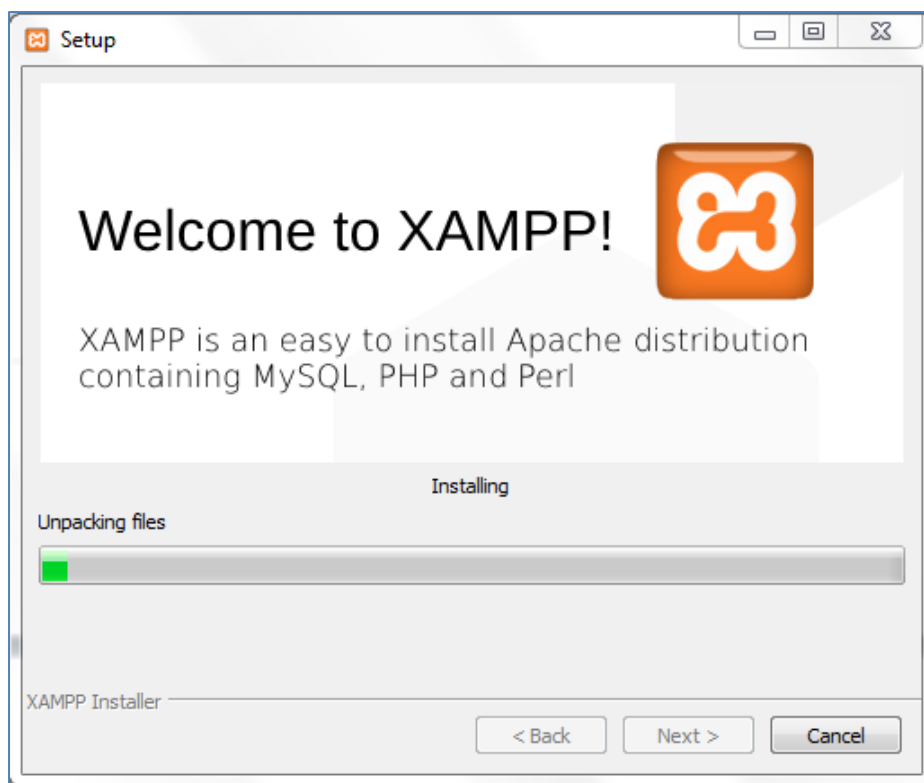


Η εγκατάσταση είναι απλή και τετριμμένη. Ακολουθώντας τις οδηγίες, όπως αυτές εμφανίζονται στην οθόνη μας και κάνοντας κλικ στο εικονίδιο Next κάθε φορά, μεταβαίνουμε από το ένα στάδιο της εγκατάστασης στο άλλο.

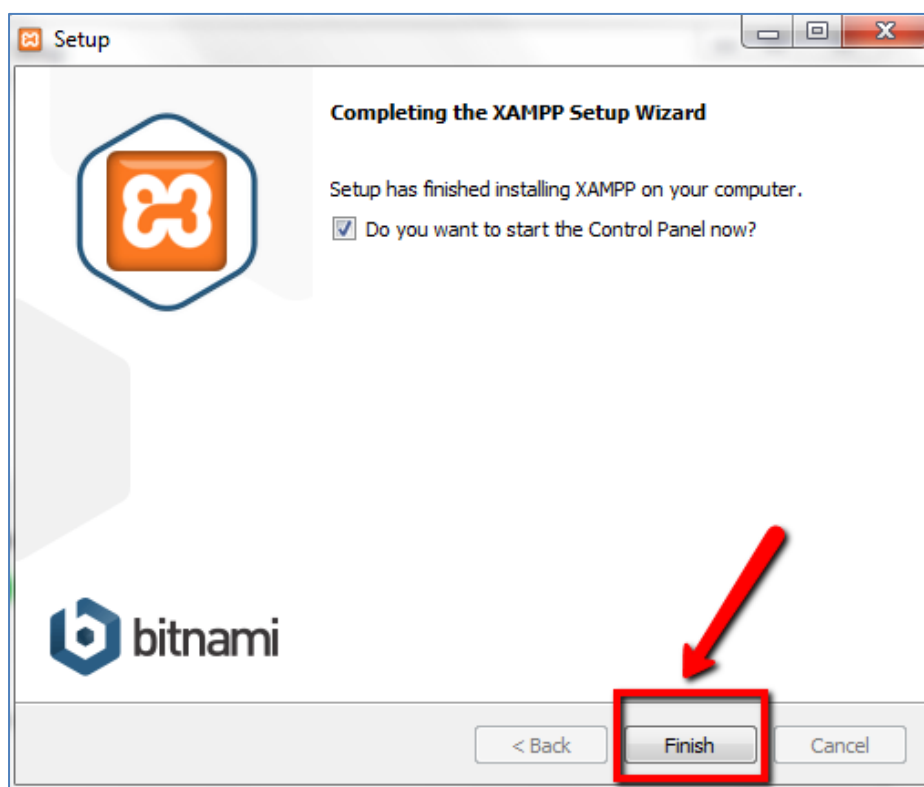


Συνεχίζεται η εγκατάσταση του xampp, όπως παρατηρείτε στις επόμενες οθόνες σας.



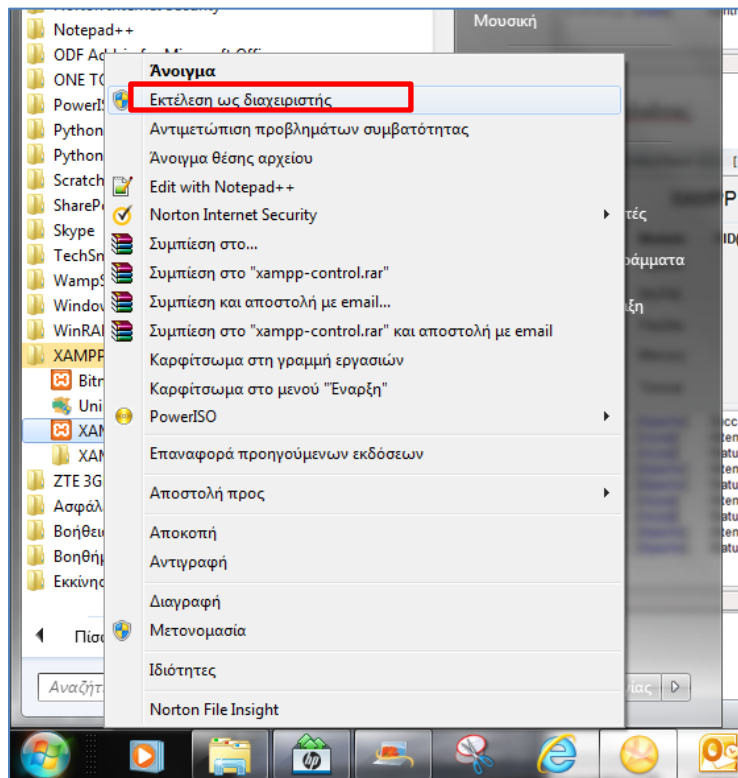


Η εγκατάσταση έχει ολοκληρωθεί. Κάνουμε κλικ στο εικονίδιο «Finish».



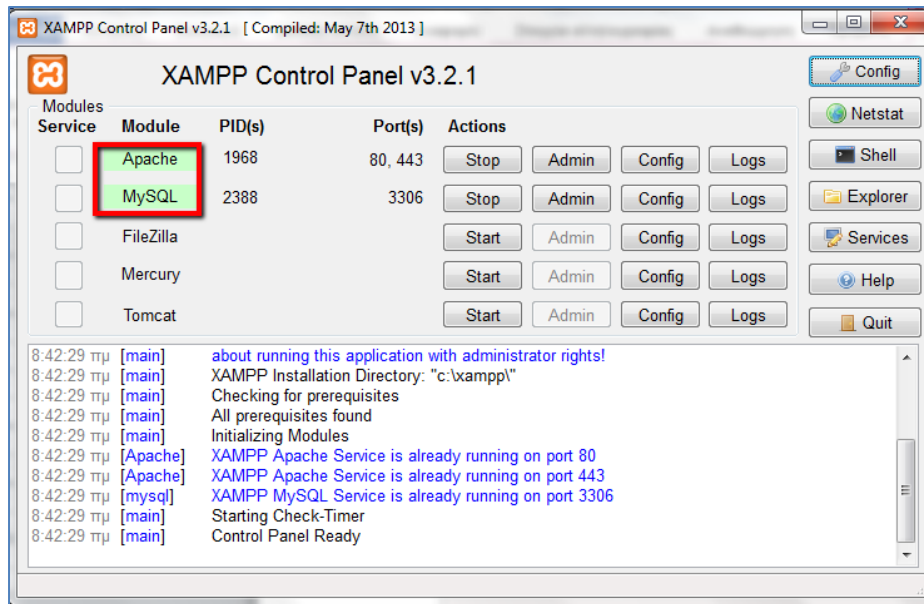
Επιλέγοντας προηγουμένως την επιλογή «Do you want to start the Control Panel now?» με την ολοκλήρωση της εγκατάστασης ξεκινάει το πρόγραμμα Control Panel για να ξεκινήσουμε/σταματήσουμε τους διακομιστές ή να εγκαταστήσουμε/

απεγκαταστήσουμε υπηρεσίες). Πατάμε start στα module του Apache και της MySql. Αν υπάρχει πρόβλημα στην ενεργοποίηση των υπηρεσιών δοκιμάστε να το εκτελέσετε το πρόγραμμα Control Panel ως διαχειριστές. Στην περίπτωση αυτή, μπορείτε να βρείτε το πρόγραμμα XAMPP Control Panel από το εικονίδιο της έναρξης και να κάνετε δεξί κλικ σε αυτό επιλέγοντας την επιλογή «Εκτέλεση ως διαχειριστής».

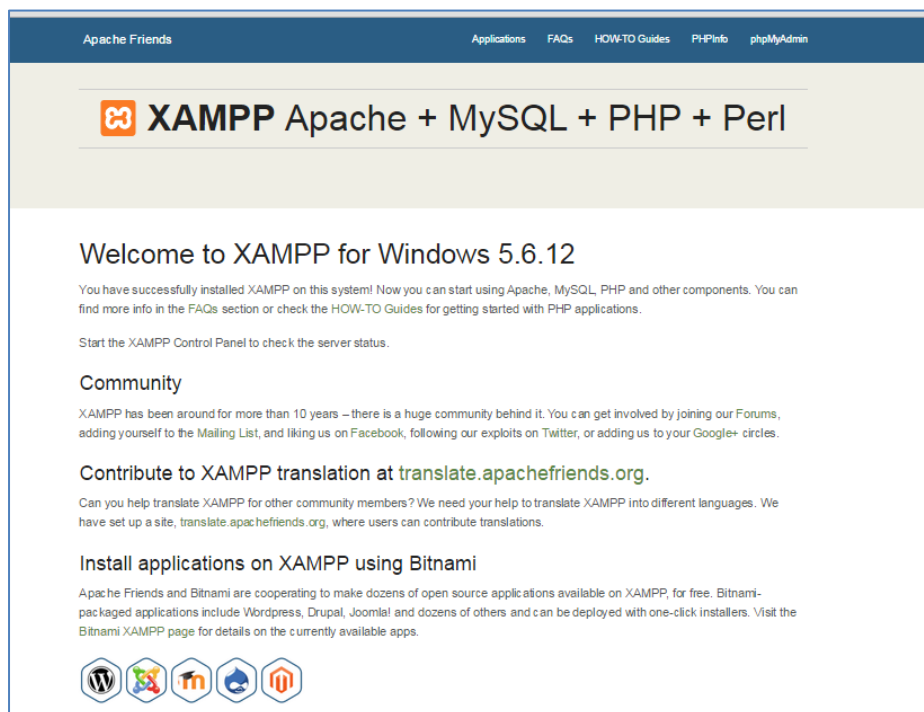


Η οθόνη του XAMPP Control Panel εμφανίζεται στην παρακάτω εικόνα. Μην ξεχνάτε να τρέχετε κάθε φορά το XAMPP Control Panel και να ενεργοποιείτε τα module του Apache και της MySql πριν ξεκινήσετε να εκτελείτε τις PHP εφαρμογές σας.

Όπως παρατηρείτε στην παρακάτω εικόνα τα module Apache και MySQL έχουν αποκτήσει πράσινο χρώμα. Αν δεν συμβαίνει αυτό προσπαθήστε να εξετάσετε αν είναι κατειλημμένη η αντίστοιχη θύρα από κάποιο άλλο πρόγραμμα, όπως είναι το skype που χρησιμοποιεί και αυτό την θύρα (port) 80. Στην περίπτωση αυτή θα πρέπει να κλείσουμε όποιο άλλο πρόγραμμα χρησιμοποιεί την ίδια θύρα ή να αλλάξουμε την θύρα στην οποία “τρέχει” ο Apache.



Για να βεβαιωθούμε ότι όλα πήγαν καλά, πληκτρολογείστε στο πρόγραμμα περιήγησής σας την διεύθυνση <http://localhost> ή <http://127.0.0.1> . Θα πρέπει να δείτε την παρακάτω εικόνα στην οθόνη σας.



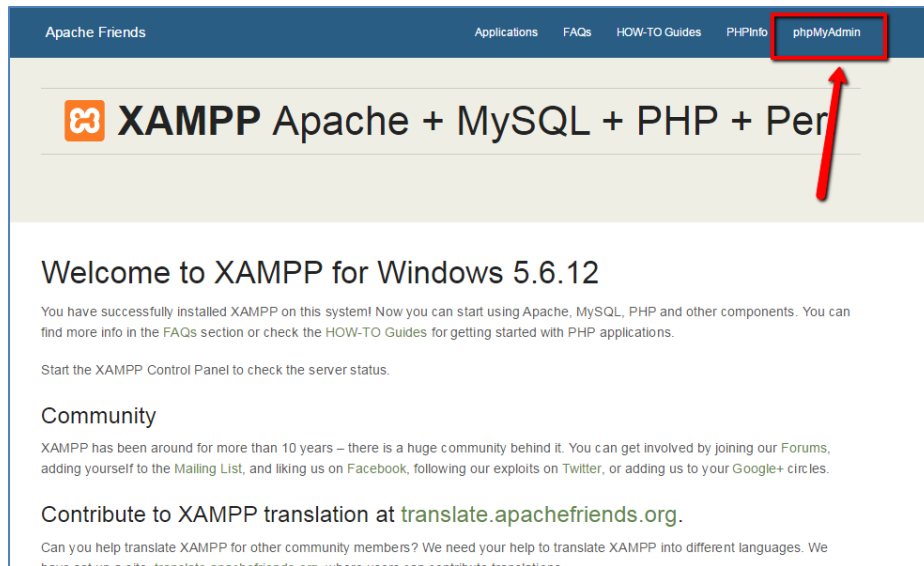
Μεταβείτε στην καρτέλα PHPInfo για να δείτε την έκδοση PHP που έχετε εγκαταστήσει.



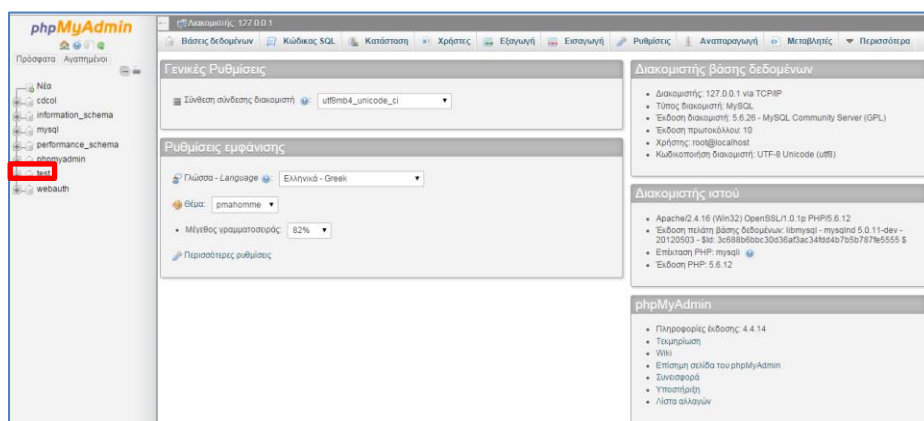
Η έκδοση PHP είναι η 5.6.12.

PHP Version 5.6.12	
System	
Build Date	Aug 6 2015 11:58:38
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x86
Configure Command	<pre> cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pdweb" "--with-pdo-oci=c:\php-sdk\oracle\instantclient_12_1\sdk\shared" "--enable-object-out-dir=.obj" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo" </pre>
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,TS,VC11
PHP Extension Build	API20131226,TS,VC11
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, sslv3, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	convert.iconv.*, mcrypt.*, mdecrypt.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*, bzip2.*

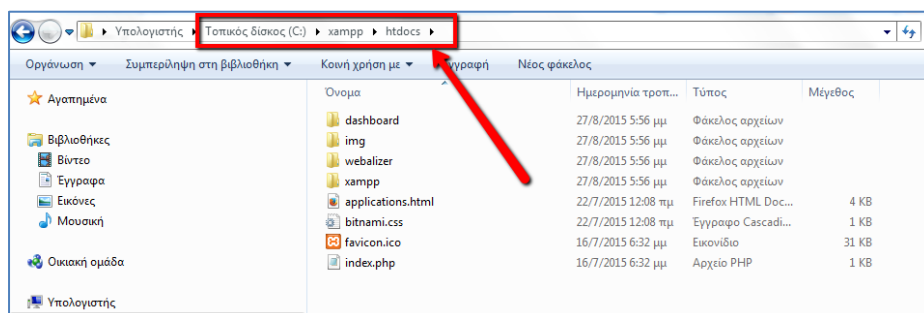
Μεταβείτε στην καρτέλα phpMyAdmin για να διαχειριστείτε την βάση δεδομένων σας.



Μέσα από το περιβάλλον της phpmyadmin μπορείτε να διαχειριστείτε την βάση δεδομένων σας. Ορισμένα ενδεικτικά παραδείγματα διαχείρισης της MySQL παρουσιάζονται σε επόμενη ενότητα. Παρατηρείστε ότι με την εγκατάσταση του xampp δημιουργήθηκε μία βάση δεδομένων με το όνομα test. Η βάση αυτή δεδομένων θα χρησιμοποιηθεί στη συνέχεια για τα πρώτα προγράμματα που θα φτιάξουμε πριν δημιουργήσουμε την δική μας βάση δεδομένων.

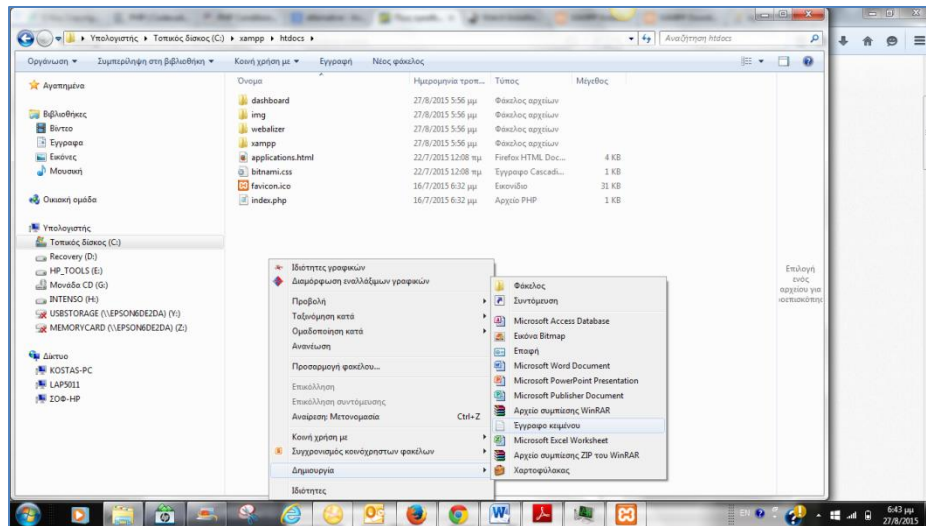


Μεταβείτε στον φάκελο c:\xampp\htdocs για να δημιουργήσετε το πρώτο πρόγραμμα σε php.

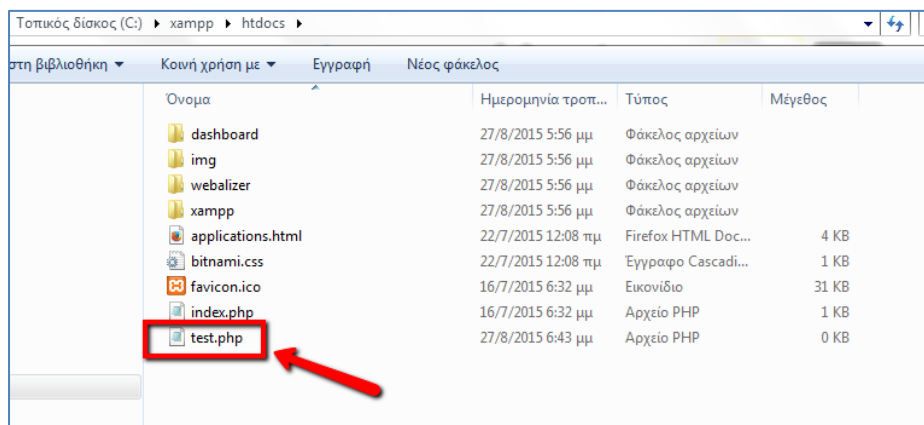




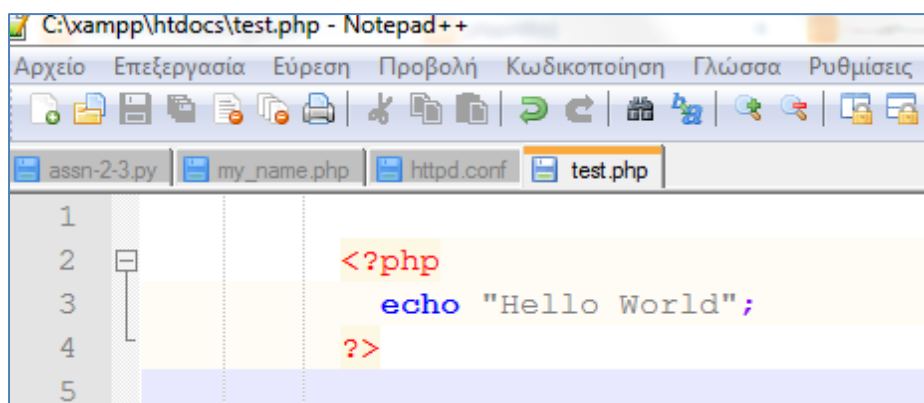
Στον φάκελο αυτό δημιουργούμε ένα αρχείο κειμένου και το ονομάζουμε test.php όπως φαίνεται στην παρακάτω οθόνη. Καλό είναι να δημιουργήσουμε έναν επιπλέον φάκελο μέσα στον φάκελο htdocs και να τοποθετήσουμε εκεί όλα τα αρχεία μας.



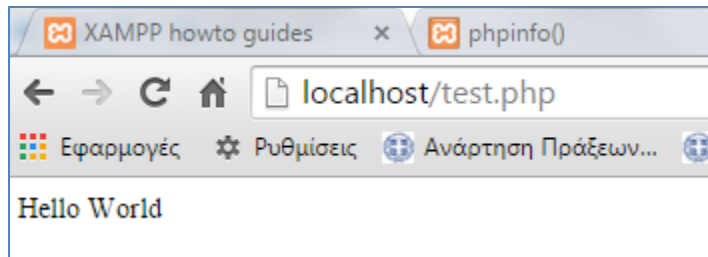
Το αρχείο test.php έχει δημιουργηθεί.



Με έναν editor για παράδειγμα με το Notepad++ πληκτρολογείτε τον ακόλουθο κώδικα, όπως φαίνεται στην παρακάτω οθόνη.



Στο πρόγραμμα περιήγησή σας πληκτρολογήστε την διεύθυνση <http://localhost/test.php> για να εκτελέσετε το πρόγραμμα test.php. Τι παρατηρείτε; Εμφανίστηκε το μήνυμα «Hello World»;



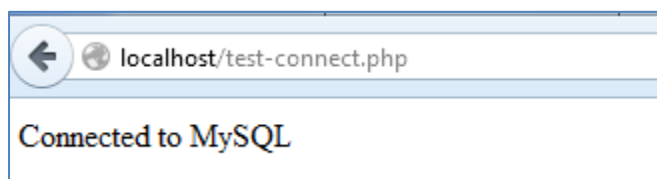
Για να σιγουρευτούμε ότι η PHP συνδέεται με τον διακομιστή της MySQL δημιουργήστε ένα νέο αρχείο με το όνομα test-connect.php στον φάκελο c:\xampp\htdocs. Ο κώδικας που θα περιέχει αυτό το αρχείο παρουσιάζεται στην συνέχεια. Χρησιμοποιείται η συνάρτηση mysql\_connect για την σύνδεση με τον διακομιστή της MySQL.

```
<?php
mysql_connect("localhost", "root", "") or die(mysql_error());

echo "Connected to MySQL<br />";

?>
```

Στο πρόγραμμα περιήγησή σας πληκτρολογήστε την διεύθυνση <http://localhost/test-connect.php> για να εκτελέσετε το πρόγραμμα test-connect.php. Τι παρατηρείτε; Εμφανίστηκε το μήνυμα “ Connected to MySQL”;



Η συνάρτηση **mysql\_connect** δέχεται τρεις παραμέτρους: το όνομα του διακομιστή (server), το όνομα του χρήστη (username) και τον κωδικό του χρήστη (password). Στο παράδειγμά μας, οι τρεις αυτοί παράμετροι έχουν τις ακόλουθες τιμές:

- **server** = “ localhost” (Οι διακομιστές PHP και MySQL είναι εγκατεστημένοι στον ίδιο τοπικό υπολογιστή)
- **username** = “root”
- **password** = “ ” (δεν έχουμε βάλει κωδικό πρόσβασης για τον root)

Η συνάρτηση die(mysql\_error()) εμφανίζει ένα μήνυμα λάθους στο πρόγραμμα περιήγησή σας στην περίπτωση που παρουσιαστεί πρόβλημα κατά την προσπάθεια σύνδεσής σας με τον διακομιστή της MySQL.

## Διαχείριση βάσεων δεδομένων μέσω PHP

Μετά τη δημιουργία σύνδεσης με τον διακομιστή της MySQL ας δημιουργήσουμε κάποια απλά προγράμματα PHP τα οποία θα «συνομιλούν» με την MySQL. Ας ξεκινήσουμε από ένα πρόγραμμα με το οποίο θα επιλέγουμε ποια βάση δεδομένων θα χρησιμοποιούμε με τη σύνδεση αυτή της MySQL. Για τον σκοπό αυτό θα χρησιμοποιήσουμε την συνάρτηση **mysql\_select\_db**, όπως φαίνεται στον παρακάτω κώδικα. Θα προσπαθήσουμε να συνδεθούμε με την βάση δεδομένων **test** που προϋπάρχει με την εγκατάσταση του xampp. Δημιουργήστε ένα νέο αρχείο με το όνομα test-connect-db.php στον φάκελο c:\xampp\htdocs και πληκτρολογήστε τον παρακάτω κώδικα.

```
<?php

mysql_connect("localhost", "root", "") or die(mysql_error());

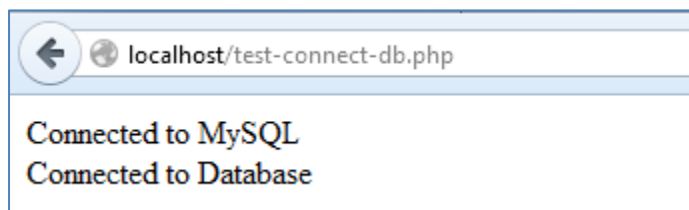
echo "Connected to MySQL<br />";

mysql_select_db("test") or die(mysql_error());

echo "Connected to Database";

?>
```

Στο πρόγραμμα περιήγησή σας πληκτρολογήστε την διεύθυνση <http://localhost/test-connect-db.php> για να εκτελέσετε το πρόγραμμα test-connect-db.php. Τι παρατηρείτε; Εμφανίστηκαν τα μηνύματα σύνδεσης με την MySQL και την βάση δεδομένων «test»;



Ας δημιουργήσουμε έναν πίνακα με το όνομα example στην βάση δεδομένων test. Θα χρησιμοποιήσουμε την συνάρτηση **mysql\_query** για να πούμε στην MySQL ότι θέλουμε να κάνει κάτι για εμάς. Στέλνουμε δηλαδή ένα ερώτημα (query). Αυτό το κάτι είναι να δημιουργήσει έναν πίνακα (με την εντολή **CREATE TABLE**) με το όνομα «example». Ο πίνακας αυτός θα περιέχει μόνο μία στήλη με το όνομα message. Στον πίνακα μπορούμε να αποθηκεύσουμε «φράσεις». Αν μετρήσουμε τους χαρακτήρες σε κάθε «φράση» που θα καταχωρούμε δεν θα πρέπει να ξεπερνάμε τους 30. Δημιουργήστε ένα νέο αρχείο με το όνομα test-create-table.php στον φάκελο c:\xampp\htdocs και πληκτρολογήστε τον παρακάτω κώδικα.

```

<?php

// Σύνδεση με την MySQL

mysql_connect("localhost", "root", "") or die(mysql_error());

mysql_select_db("test") or die(mysql_error());

// Δημιουργία πίνακα στην επιλεγμένη βάση “test”

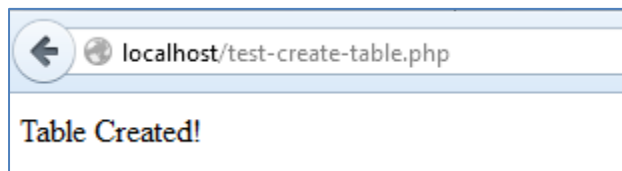
mysql_query("CREATE TABLE example(message VARCHAR(30))") or
die(mysql_error());

echo "Table Created!";

?>

```

Στο πρόγραμμα περιήγησή σας πληκτρολογήστε την διεύθυνση <http://localhost/test-create-table.php> για να εκτελέσετε το πρόγραμμα test-create-table.php. Τι παρατηρείτε; Εμφανίστηκε το μήνυμα δημιουργίας πίνακα;



Στη συνέχεια, ας προσπαθήσουμε να εισάγουμε κάποιες πληροφορίες/φράσεις στον πίνακά μας και συγκεκριμένα την φράση «I like solving problems». Χρησιμοποιούμε την συνάρτηση mysql\_query. Η εντολή **INSERT INTO** σημαίνει ότι θέλουμε να καταχωρήσουμε δεδομένα σε έναν πίνακα και συγκεκριμένα στον πίνακα «example». Δημιουργήστε ένα νέο αρχείο με το όνομα test-insert-data.php στον φάκελο c:\xampp\htdocs και πληκτρολογήστε τον παρακάτω κώδικα.

```

<?php

// Σύνδεση με την MySQL

mysql_connect("localhost", "root", "") or die(mysql_error());

mysql_select_db("test") or die(mysql_error());

// Εισαγωγή μιας εγγραφής στον πίνακα “example”

mysql_query("INSERT INTO example
(message) VALUES('I like solving problems') ")

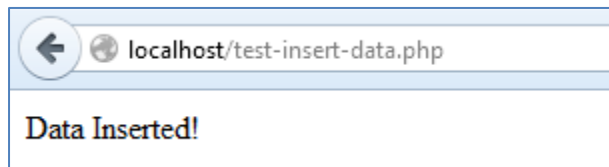
```

```
or die(mysql_error());

echo "Data Inserted!";

?>
```

Στο πρόγραμμα περιήγησή σας πληκτρολογήστε την διεύθυνση <http://localhost/test-insert-data.php> για να εκτελέσετε το πρόγραμμα test-insert-data.php. Τι παρατηρείτε; Εμφανίστηκε το μήνυμα εισαγωγής δεδομένων;



Ας εμφανίσουμε στην οθόνη μας την φράση «I like solving problems» που μόλις καταχωρήσαμε στον πίνακά μας. Χρησιμοποιούμε και εδώ την συνάρτηση `mysql_query`. Με την εντολή **SELECT** ζητάμε από την MySQL να μας στείλει τα περιεχόμενα του πίνακα `example` (στην συγκεκριμένη περίπτωση για λόγους απλότητας υπάρχει καταχωρημένη μόνο μία φράση). Για να μπορέσουμε τα περιεχόμενα αυτά να τα χρησιμοποιήσουμε (ή να τα εμφανίσουμε) αργότερα μέσα στο PHP πρόγραμμά μας, τα αποθηκεύουμε στην μεταβλητή `$result`.

Η συνάρτηση **`mysql_fetch_array`** μας επιστρέφει το πρώτο αποτέλεσμα από το ερώτημα `SELECT` με την μορφή πίνακα. Με την βοήθεια της PHP συνάρτησης `mysql_fetch_array` μπορούμε να εμφανίσουμε το περιεχόμενο του πίνακα `example` στην επιθυμητή μορφή. Σε περίπτωση, που είχαμε περισσότερα που ένα αποτελέσματα έπρεπε να τοποθετήσουμε την συνάρτηση `mysql_fetch_array` μέσα σε μία δομή επανάληψης για να μπορέσουμε να τα εμφανίσουμε όλα.

Δημιουργήστε ένα νέο αρχείο με το όνομα `test-quey-db.php` στον φάκελο `c:\xampp\htdocs` και πληκτρολογήστε τον παρακάτω κώδικα.

```
<?php

// Σύνδεση με την MySQL

mysql_connect("localhost", "root", "") or die(mysql_error());

mysql_select_db("test") or die(mysql_error());

// Ανάκτηση των δεδομένων από τον πίνακα "example"

$result = mysql_query("SELECT * FROM example")

or die(mysql_error());
```

```
// Αποθήκευση της πρώτης εγγραφής του πίνακα "example" στην μεταβλητή $row
//που έχει την μορφή πίνακα

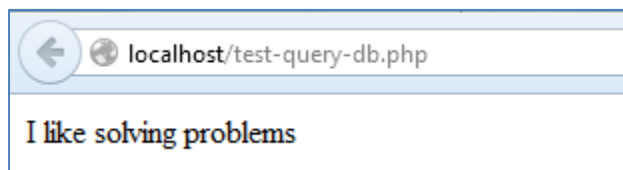
$row = mysql_fetch_array( $result );

// Εμφάνιση της τιμής της μεταβλητής $row

echo $row['message'];

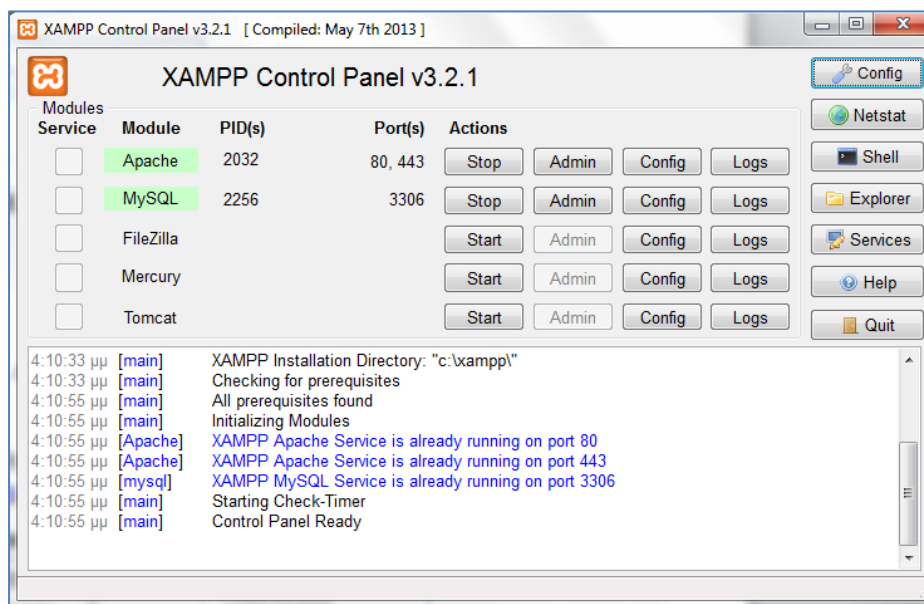
?>
```

Στο πρόγραμμα περιήγησή σας πληκτρολογήστε την διεύθυνση <http://localhost/test-query-db.php> για να εκτελέσετε το πρόγραμμα test-query-db.php. Τι παρατηρείτε; Εμφανίστηκε το περιεχόμενο του πίνακα «example»;

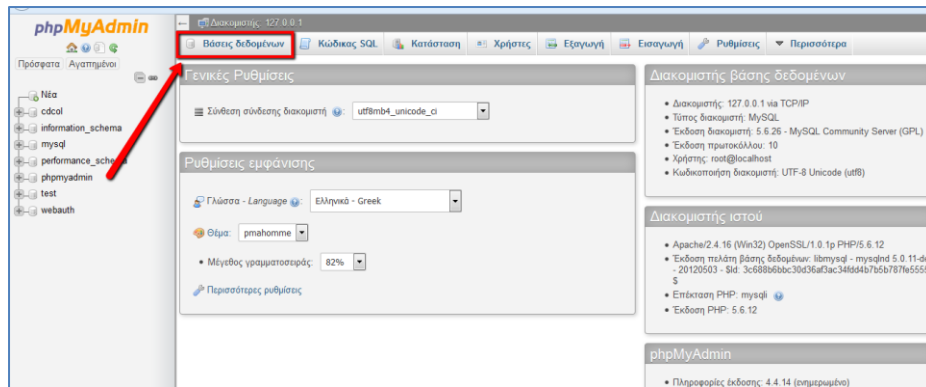


### Δημιουργία βάσης δεδομένων με phpmyadmin

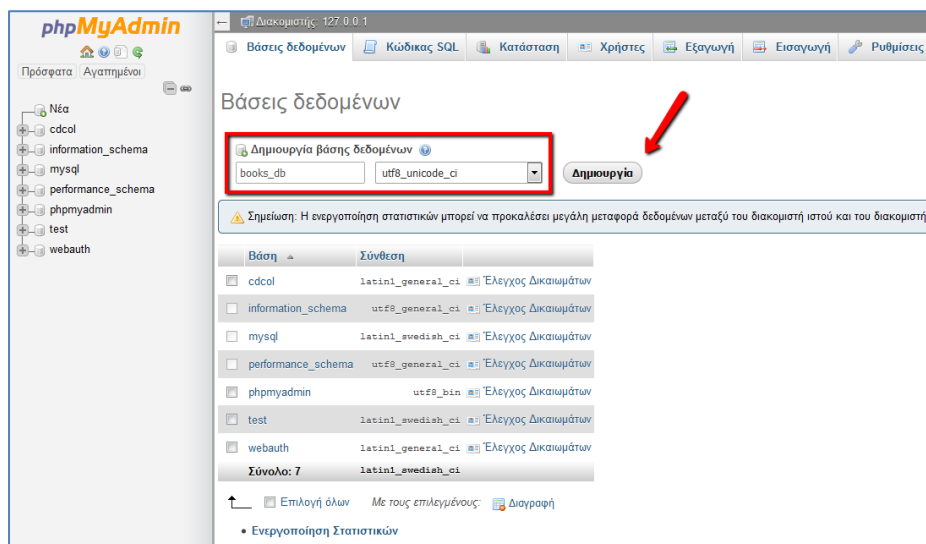
Εκκινήστε το Control Panel του XAMPP και ενεργοποιήστε τις υπηρεσίες Apache και MySQL αν χρειάζεται.



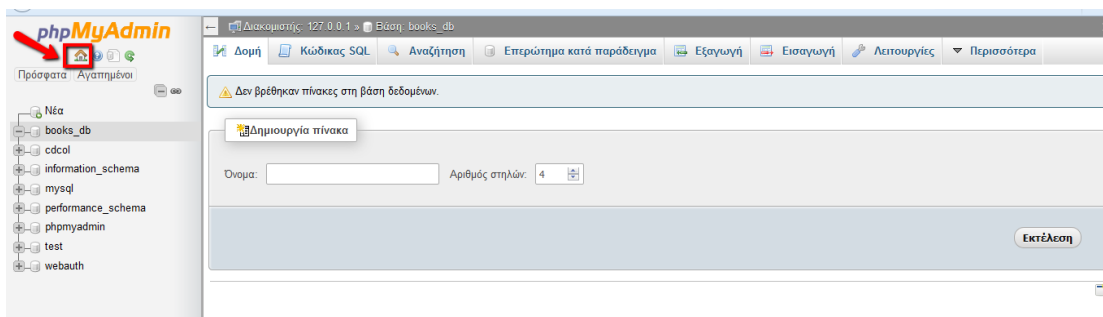
Πληκτρολογήστε <http://localhost/phpmyadmin> και κάντε κλικ στην καρτέλα **Βάσεις δεδομένων**.



Πληκτρολογήστε το όνομα της Βάσης στο σχετικό πεδίο και από το μενού **Σύνδεση** **Σύνθεσης Διακομιστή** επιλέξτε **utf8\_unicode\_ci**. Πατήστε «**Δημιουργία**».

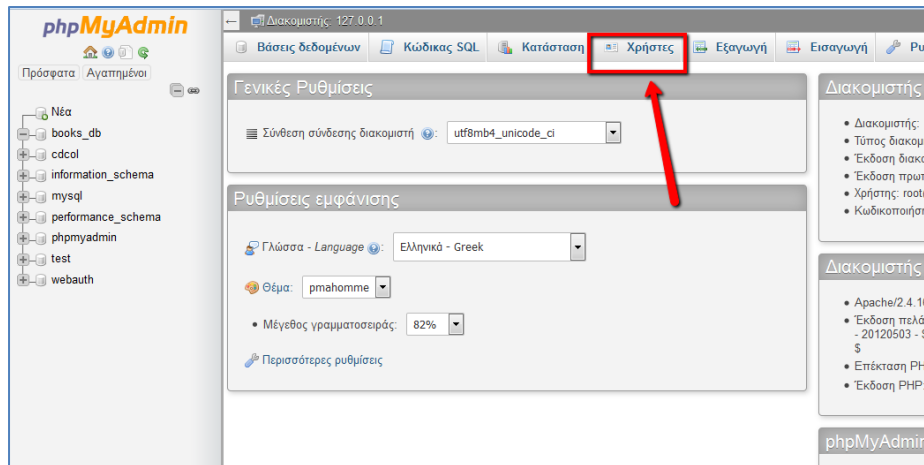


Δημιουργήθηκε η βάση μας. Στο σημείο αυτό αν θέλετε μπορείτε να δημιουργήσετε πίνακα. Ας μεταβούμε στην αρχική καρτέλα κάνοντας κλικ στο αντίστοιχο εικονίδιο, όπως φαίνεται στην παρακάτω εικόνα.

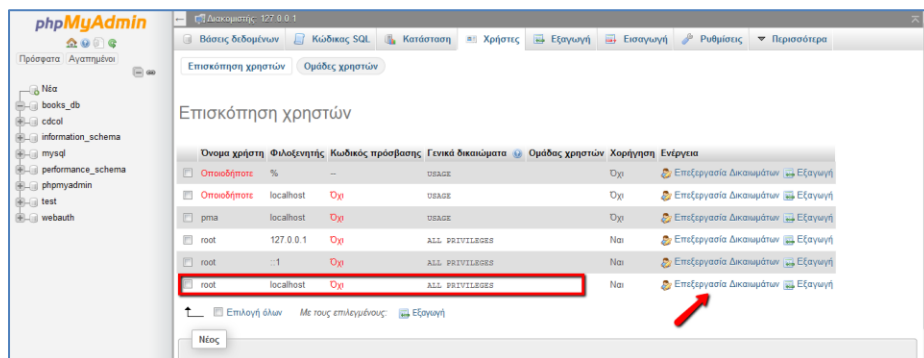


Ας αλλάξουμε το password του «**root**» της MySQL για λόγους ασφαλείας, όπου η default ρύθμιση είναι χωρίς κωδικό. Μεταβείτε στην καρτέλα **Χρήστες**.

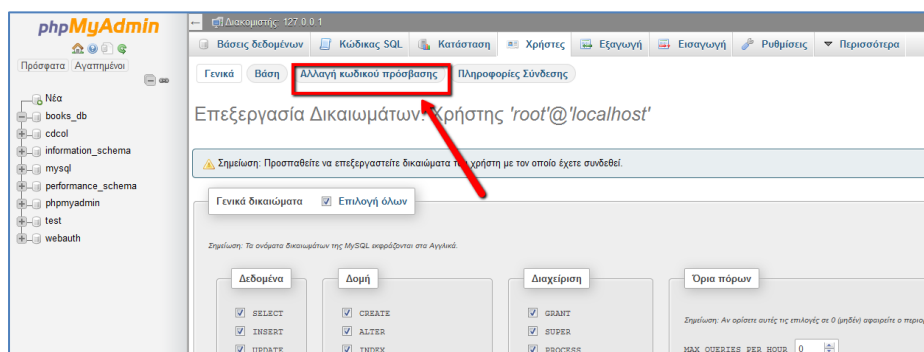
## ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ



Επιλέγουμε τον χρήστη “root” όπως φαίνεται στην παρακάτω εικόνα και κάνουμε κλικ στην επιλογή στο «Επεξεργασία Δικαιωμάτων».

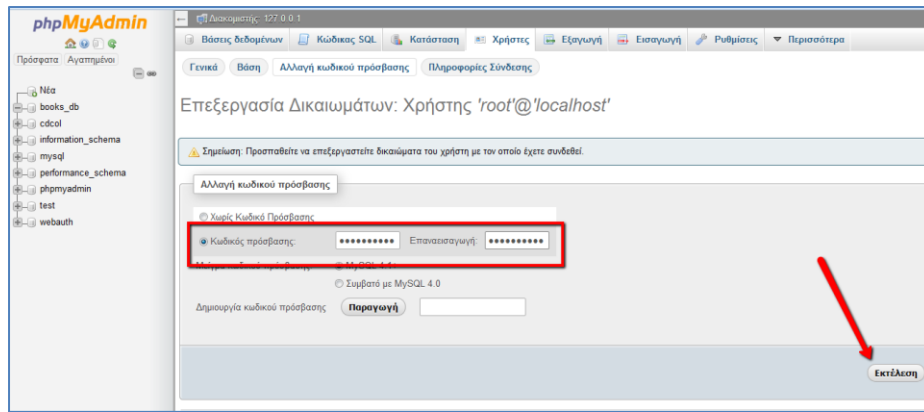


Στη συνέχεια επιλέγουμε “Αλλαγή κωδικού πρόσβασης”.

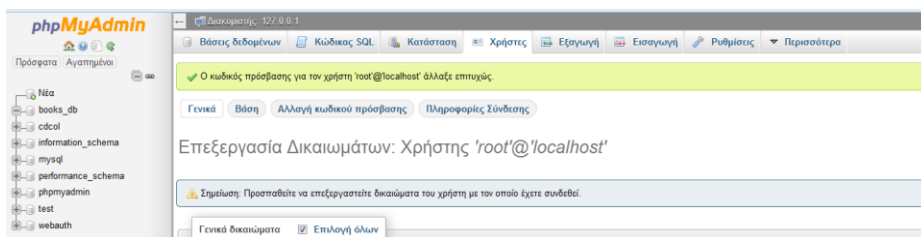


Ορίζουμε τον επιθυμητό κωδικό, για παράδειγμα «mypassword» για λόγους απλότητας και κάνουμε κλικ στο «Εκτέλεση».





Η αλλαγή του κωδικού έγινε με επιτυχία.



Τέλος, θα πρέπει να ορίσουμε το νέο κωδικό στο αρχείο config.inc.php (που βρίσκεται στον φάκελο C:\xampp\phpMyAdmin) στη παρακάτω σειρά:

```
$cfg['Servers'][$i]['password'] = 'mypassword';
```

Αν θελήσουμε να εκτελέσουμε, όπως και στην αρχή της προηγούμενης ενότητας, ένα πρόγραμμα PHP σύνδεσης με τον διακομιστή της MySQL θα πρέπει να πληκτρολογήσουμε τον παρακάτω κώδικα δίνοντας ως username το «root» και ως password το «mypassword».

```
<?php
mysql_connect("localhost", "root", "mypassword") or die(mysql_error());

echo "Connected to MySQL<br />";

?>
```

### Διαδικτυακές Πηγές

- Εγκατάσταση Apache, php, MySQL, phpMyAdmin: ΚΕ.ΠΛΗ.NET Δυτικής Θεσσαλονίκης  
<http://dide-v.thess.sch.gr/plinet/install-apache-php-mysql-phpmyadmin>
- Εγκατάσταση xampp: Πανελλήνιο Σχολικό Δίκτυο  
<http://www.sch.gr/3187-server#>